

## DANE OSOBOWE

---

*Data i miejsce urodzenia* 6 grudnia 1990, Gliwice  
*Adres* Wydział Matematyki, Informatyki i Mechaniki  
Uniwersytetu Warszawskiego  
ul. Banacha 2, 02-097 Warszawa  
*E-Mail* kociumaka@mimuw.edu.pl  
*Strona domowa* <https://www.mimuw.edu.pl/~kociumaka>

## WYKSZTAŁCENIE

---

**Informatyka, studia III stopnia** 2014–

*Uniwersytet Warszawski; Wydział Matematyki, Informatyki i Mechaniki*

Promotor: prof. dr hab. Wojciech Rytter

Tytuł pracy: Efektywne struktury danych dla zapytań wewnętrznych w tekstach

**Informatyka, studia II stopnia** 2012–2014

*Uniwersytet Warszawski; Wydział Matematyki, Informatyki i Mechaniki*

Promotor: dr Marek Cygan

Tytuł pracy: Algorytmy aproksymacyjne dla problemów lokalizacji obiektów z pojemnościami

**Informatyka, studia I stopnia** 2009–2012

*Uniwersytet Warszawski; Wydział Matematyki, Informatyki i Mechaniki*

## NAGRODY I WYRÓŻNIENIA

---

**LATA 2018 Best Paper Award** 2018

*12th International Conference on Language and Automata Theory and Applications*

wspólnie z J. Radoszewskim, W. Rytterem i T. Waleniem

**CPM 2016 Alberto Apostolico Best Paper Award** 2016

*27th Annual Symposium on Combinatorial Pattern Matching*

**Stypendium Ministra Nauki i Szkolnictwa Wyższego dla doktorantów** 2016

**Konkurs na najlepsze prace magisterskie z zakresu informatyki i jej zastosowań** 2015

I nagroda, *Polskie Towarzystwo Informatyczne*

**Stypendium Ministra Nauki i Szkolnictwa Wyższego dla studentów** 2010–2013

## KIEROWNICTWO GRANTÓW

---

**Algorytmiczne i kombinatoryczne aspekty okresowości i powtórzeń w słowach** 2013–2017

Diamentowy Grant, *Ministerstwo Nauki i Szkolnictwa Wyższego*

## NAGRODY W KONKURSACH

---

<i>Google Code Jam</i>	2015
finalista, 7. miejsce	
<i>Facebook Hacker Cup</i>	2015
finalista	
<i>ACM International Collegiate Programming Contest</i>	2013–2014
srebrny medal (dwukrotnie): 2013 — 6. miejsce, 2014 — 5. miejsce	
<i>International Mathematical Competition (IMC)</i>	2010–2013
nagroda I stopnia (czterokrotnie), 2012 — 2. miejsce	
<i>Akademickie Mistrzostwa Polski w Programowaniu Zespołowym</i>	2010–2013
2010 — 3. miejsce, 2011 i 2012 — 2. miejsce, 2013 — 1. miejsce	
<i>21. Międzynarodowa Olimpiada Informatyczna</i>	2009
złoty medal, 7. miejsce	
<i>50. Międzynarodowa Olimpiada Matematyczna</i>	2009
srebrny medal	
<i>XVI Olimpiada Informatyczna</i>	2009
1. miejsce	
<i>LX Olimpiada Matematyczna</i>	2009
2. miejsce	

## ZAINTERESOWANIA NAUKOWE

---

Od początku mojej pracy naukowej, czyli od połowy studiów licencjackich, główny nurt prowadzonych przeze mnie badań związany jest z algorytmiką tekstów, czyli opracowywaniem wydajnych metod przetwarzania tekstów rozumianych jako dowolne, abstrakcyjne ciągi znaków. Poniżej przedstawiam cztery najważniejsze kierunki mojej dotychczasowej pracy w tej dziedzinie.

Okazjonalnie zajmowałem się także zagadnieniami z innych działów algorytmiki. Moja praca magisterska dotyczy algorytmów aproksymacyjnych, a tematy poruszane na przedmiotach monograficznych zainspirowały mnie do badań zwieńczonych kilkoma publikacjami w dziedzinie algorytmów parametryzowanych i umiarkowanie wykładniczych oraz jedną dotyczącą algorytmów rozproszonych.

### Struktury danych do zapytań wewnętrznych

Cykl prac [35, 23, 3, 4, 18] przedstawiających efektywne struktury danych dla tzw. *zapytań wewnętrznych* w tekstach stanowi podstawę mojej rozprawy doktorskiej. W rozważanym modelu po przetworzeniu tekstu można zadawać pytania dotyczące jego fragmentów (identyfikowanych przez pozycje początkową i końcową). Fundamentalne problemy tego rodzaju to sprawdzanie równości fragmentów, ich porównywanie w porządku leksykograficznym oraz wyznaczanie ich najdłuższego wspólnego prefiksu.

Pierwszym obiektem moich badań, prowadzonych w zespole na UW, były zapytania o zbiór okresów wskazanego fragmentu [35]. Jednym z wyników kolejnej publikacji [3], zaprezentowanej na konferencji SODA 2015, jest bardziej efektywna struktura danych wielkości  $O(n)$  (dla tekstu długości  $n$ ), która wyznacza okresy w czasie  $O(\log n)$ , a ponadto może być skonstruowana za pomocą algorytmu randomizowanego w oczekiwanym czasie liniowym (w rozprawie przedstawiam wersję deterministyczną). Podstawowym problemem rozważanym w [3] jest jednak wyszukiwanie wzorca, tzn. pytania o lokalizację wystąpień jednego fragmentu wewnątrz innego fragmentu, na które ta sama struktura danych odpowiada w czasie proporcjonalnym do ilorazu długości rozważanych fragmentów.

W rozprawie przedstawiam także opartą na zbliżonych pomysłach strukturę do zapytań o najdłuższy wspólny prefiks fragmentów tekstu nad małym alfabetem (wielkości  $\sigma \ll n$ ). Odpowiada ona na zapytania w czasie stałym i można ją skonstruować w czasie  $O(n/\log_\sigma n)$ , korzystając z faktu, że słowo maszynowe modelu RAM mieści  $\Omega(\log_\sigma n)$  znaków. Klasyczne odpowiedniki wymagają czasu konstrukcji  $\Theta(n)$  nawet wtedy, gdy celem jest jedynie sprawdzanie równości fragmentów.

W innym zespole zajmowaliśmy się natomiast zapytaniami o najmniejszy oraz największy leksykograficznie sufiks fragmentu. Problem ten został wprowadzony przez Babenkę i in., którzy opisali strukturę danych wielkości liniowej odpowiadającą na zapytania w czasie  $O(\log n)$  (o największy sufiks)

lub  $O(\log^{1+\varepsilon} n)$  (o najmniejszy sufix). Rok później, wspólnie z autorami wspomnianej pracy przedstawiliśmy rozwiązania uzyskujące w obu przypadkach stały czas zapytania, jednak czas konstrukcji struktury dla problemu najmniejszego sufixu wynosił  $O(n \log n)$ . (Wspólna wersja czasopismowa obydwu ww. prac to [23]). W pełni optymalne rozwiązanie udało mi się uzyskać dopiero za pomocą istotnie innych technik w samodzielnej pracy [18]. Co więcej, opisana tam struktura danych pozwala w czasie stałym znajdować minimalne i maksymalne sufixy nie tylko pojedynczego fragmentu, ale również konkatenacji większej liczby fragmentów. Istotnym zastosowaniem tego uogólnionego problemu jest wyznaczanie najmniejszej leksykograficznie rotacji cyklicznej fragmentu (lub konkatenacji wielu fragmentów), a tym samym tzw. kanonizacja fragmentów ze względu na równoważność cykliczną. Wyniki te zostały wyróżnione nagrodą im. Alberto Apostolico za najlepszą pracę na konferencji CPM 2016.

Ogólniejsze zapytania rozważaliśmy natomiast w drugiej pracy [4] zaprezentowanej na konferencji SODA 2015. Badaliśmy tam problem symulacji tablicy sufixowej i odwrotnej tablicy sufixowej fragmentu, czyli zapytania o  $k$ -ty najmniejszy sufix fragmentu (dla dowolnego  $k$ ) oraz *range* wskazanego sufixu w porządku leksykograficznym. Jako efekt uboczny naszych badań uzyskaliśmy też usprawnienia dla bardziej podstawowego problemu obliczania  $k$ -tej najmniejszej liczby w spójnym fragmencie tablicy: otrzymaliśmy jednocześnie czas konstrukcji  $O(n\sqrt{\log n})$  oraz optymalny czas zapytania  $O(\log n / \log \log n)$ . Wcześniej znane były tylko struktury osiągające jeden z tych dwóch celów.

Kolejna grupa zapytań wewnętrznych dotyczy kompresji fragmentów tekstu. Pierwsze wyniki na ten temat przedstawili Cormode i Muthukrishnan na konferencji SODA 2005. Narzędzia stworzone w [3] pozwoliły nam przyspieszyć algorytm zwracający pod słowo w jednym z wariantów kompresji LZ77, natomiast w pracy [4] przedstawiliśmy strukturę danych kompresującą fragmenty tekstu za pomocą transformaty Burrowsa–Wheeler.

## Dynamiczne kolekcje tekstów

Za jeden z najbardziej wartościowych wyników uważam przedstawioną na konferencji SODA 2018 [1] strukturę danych do utrzymywania dynamicznego zbioru słów, który można rozszerzać o nowe słowo podane na wejściu, konkatenację dwóch słów ze zbioru lub podział słowa ze zbioru (na prefiks i sufix). Zapytania polegają na porównaniu leksykograficznym dwóch słów ze zbioru oraz wyznaczeniu ich najdłuższego wspólnego prefiksu. W pracy podajemy rozwiązanie, w którym operacje dzielenia i konkatenacji działają w czasie logarytmicznym (względem łącznej długości słów w zbiorze), a zapytania — w czasie stałym. Pokazujemy ponadto bezwarunkowe ograniczenie dolne, które dowodzi, że funkcja logarytmiczna jest tutaj asymptotycznie optymalna. Wyniki te poprawiają klasyczne już prace Mehlhorna i in. oraz Alstrup i in. Stworzone przez nas rozwiązania udało nam się później zaadaptować m.in. do wyszukiwania wzorca w dynamicznych tekstach oraz do przetwarzania skompresowanych tekstów bez konieczności ich dekompresji.

## Przetwarzanie tekstów w pamięci podliniowej

Dalsze wyniki dotyczą modelu, gdzie tekst jest dostępny za pomocą wyrocni (lub, równoważnie, przechowywany w pamięci tylko do odczytu), a więc złożoność pamięciowa algorytmów i rozmiar struktur danych mogą być istotnie mniejsze niż długość tekstu. Celem jest więc wykorzystywanie możliwie małej pamięci roboczej przy zachowaniu czasu działania jak najbliższego znanemu z klasycznych rozwiązań.

Mój najważniejszy wynik w tej dziedzinie to stworzone wspólnie z P. Gawrychowskim i opublikowane na konferencji SODA 2017 algorytmy konstrukcji rzadkiego drzewa sufixowego, co w szczególności wymaga leksykograficznego uporządkowania wskazanych  $b$  spośród wszystkich  $n$  sufixów danego tekstu długości  $n$ . W pracy [2] przedstawiamy dwie procedury. Nasz algorytm Monte Carlo działa w optymalnym czasie  $O(n)$  i pamięci  $O(b)$ , a przy tym jest poprawny z dużym prawdopodobieństwem (tzn. myli się z prawdopodobieństwem  $O(n^{-c})$  dla dowolnie wybranej stałej  $c > 0$ ). Deterministyczny algorytm weryfikacji, którego można użyć do wykrycia ewentualnego błędu, tj. w konstrukcji randomizowanej typu Las Vegas, wykorzystuje natomiast czas  $O(n\sqrt{\log b})$  i pamięć  $O(b)$ . Poprzednie algorytmy w pamięci  $O(b)$  dla obydwu modeli randomizacji wymagają czasu  $O(n \log b)$ .

Wcześniej, na konferencji ESA 2015 [9] w szerszym gronie autorów przedstawiliśmy algorytm wyszukiwania wielu wzorców w pamięci proporcjonalnej do ich liczby. W przeciwieństwie do poprzednich rozwiązań o zbliżonych parametrach, nasz algorytm obsługuje wzorce różnej długości. Wykorzystaliśmy

go następnie do stworzenia metody obliczania aproksymacyjnej kompresji typu LZ77 ( $1 + \varepsilon$  razy większej niż optymalna) w pamięci roboczej proporcjonalnej do skompresowanej reprezentacji tekstu.

W innym zespole uzyskaliśmy także nowe wyniki dla problemu najdłuższego wspólnego pod słowa tekstów [10]: algorytm działający w czasie  $O(n\tau)$  i pamięci  $O(\frac{n}{\tau})$  dla dowolnie wybranego parametru  $\tau$ .

## Algorytmika i kombinatoryka nieklasycznych tekstów

Kolejny kierunek moich badań dotyczy tzw. nieklasycznych tekstów, w których pasujące do siebie słowa nie muszą być równe, lecz relacja ta zdefiniowana jest w niestandardowy, mniej restrykcyjny sposób.

W modelu abelowym słowa uznaje się za równoważne, jeśli są anagramami, tzn. jedno można przekształcić w drugie, zmieniając kolejność liter. W zespole na UW opracowaliśmy algorytmy wyznaczania okresów abelowych [43, 7] oraz indeks (strukturę danych do wyszukiwania fragmentów tekstu pasujących do wzorca) dla tekstów nad alfabetem stałej wielkości [11]. Jest to pierwsza tego rodzaju struktura danych, która dla alfabetu większego niż binarny osiąga rozmiar silnie podkwadratowy (tzn.  $O(n^{2-\Omega(1)})$ ) dla tekstu długości  $n$  oraz czas zapytania silnie podliniowy (tzn.  $O(n^{1-\Omega(1)})$ ). Chan i Lewenstein opracowali później silnie podkwadratowy algorytm konstrukcji takiego indeksu (STOC 2015).

Kolejne modele nieklasycznych tekstów to równość parametryzowana, gdzie słowa są równoważne, jeśli jedno da się przekształcić na drugie permutując alfabet (zmieniając nazwy liter), oraz równość kształtów, gdzie każdy symbol ma wartość liczbową, a przekształcenie liter jednego słowa na litery drugiego musi zachować porządek tych wartości. Cole i Hariharan przedstawili na konferencji STOC 1997 efektywny randomizowany algorytm konstrukcji drzewa sufikсового w modelu parametryzowanym: ich procedura z dużym prawdopodobieństwem działa w czasie liniowym. Ten sam algorytm można zastosować w modelu równości kształtów, ale czas działania zwiększa się do  $O(n \log n / \log \log n)$ . W pracy [13] opisaliśmy nieco zmodyfikowaną wersję drzewa sufikсового, którą można skonstruować w czasie  $O(n \log \log n)$  (z dużym prawdopodobieństwem) i, podobnie jak oryginalnego drzewa, używać do wyszukiwania wzorca w czasie liniowym od jego długości. Przedstawiliśmy też liczne zastosowania naszej struktury danych, m.in. do obliczania najdłuższego wspólnego pod słowa wielu tekstów oraz do wyznaczania kwadratów w modelu równości kształtów. Narzędzia te wykorzystaliśmy później do wyznaczania okresów w tym modelu [6].

Zajmowaliśmy się także słowami z blankami (ang. *partial words*), gdzie jeden z symboli alfabetu (nazywany blankiem lub dziurą) pasuje do wszystkich symboli. Nasze wyniki w tej dziedzinie zawarte są w trzech publikacjach [44, 16, 36]. Najciekawsza z nich [36] dotyczy rozszerzenia klasycznego lematu o okresowości. Po blisko 20 latach od pierwszych badań na ten temat, podaliśmy pełną charakteryzację funkcji  $L(p, q, h)$ , której wartość zdefiniowana jest jako najmniejsza długość  $n$  słów częściowych z  $h$  blankami, w których obecność okresów  $p$  i  $q$  musi implikować okres  $\text{nwd}(p, q)$ . Wyniki te zostały wyróżnione nagrodą na najlepszą pracę na konferencji LATA 2018.

## PUBLIKACJE

---

### Algorytmy tekstowe

- [1] P. Gawrychowski, A. Karczmarz, T. Kociumaka, J. Łącki i P. Sankowski. Optimal dynamic strings. [W:] *29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*. Red. A. Czumaj. SIAM, 2018, s. 1509–1528. DOI: 10.1137/1.9781611975031.99.
- [2] P. Gawrychowski i T. Kociumaka. Sparse suffix tree construction in optimal time and space. [W:] *28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*. Red. P. Klein. SIAM, 2017, s. 425–439. DOI: 10.1137/1.9781611974782.27.
- [3] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Internal pattern matching queries in a text and applications. [W:] *26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*. Red. P. Indyk. SIAM, 2015, s. 532–551. DOI: 10.1137/1.9781611973730.36.
- [4] M. Babenko, P. Gawrychowski, T. Kociumaka i T. Starikovskaya. Wavelet trees meet suffix trees. [W:] *26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*. Red. P. Indyk. SIAM, 2015, s. 572–591. DOI: 10.1137/1.9781611973730.39.

- [5] T. Kociumaka, M. Kubica, J. Radoszewski, W. Rytter i T. Waleń. A linear time algorithm for seeds computation. [W:] *23rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012*. Red. Y. Rabani. SIAM, 2012, s. 1095–1112. DOI: 10.1137/1.9781611973099.86.
- [6] G. Gourdel, T. Kociumaka, J. Radoszewski, W. Rytter, A. Shur i T. Waleń. String periods in the order-preserving model. [W:] *Symposium on Theoretical Aspects of Computer Science, STACS 2018*. Red. R. Niedermeier i B. Vallée. T. 96. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 38:1–38:16. DOI: 10.4230/LIPIcs.STACS.2018.38.
- [7] T. Kociumaka, J. Radoszewski i W. Rytter. Fast algorithms for Abelian periods in words and greatest common divisor queries. *Journal of Computer and System Sciences* 84 (2017), s. 205–218. DOI: 10.1016/j.jcss.2016.09.003.  
Wersja konferencyjna: Fast algorithms for Abelian periods in words and greatest common divisor queries. [W:] *Symposium on Theoretical Aspects of Computer Science, STACS 2013*. Red. N. Portier i T. Wilke. T. 20. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013, s. 245–256. DOI: 10.4230/LIPIcs.STACS.2013.245.
- [8] M. Gańczorz, P. Gawrychowski, A. Jeż i T. Kociumaka. Edit distance with block operations. [W:] *Algorithms — ESA 2018*. Red. Y. Azar, H. Bast i G. Herman. T. 112. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 33:1–33:14. DOI: 10.4230/LIPIcs.ESA.2018.33.
- [9] J. Fischer, T. Gagie, P. Gawrychowski i T. Kociumaka. Approximating LZ77 via small-space multiple-pattern matching. [W:] *Algorithms — ESA 2015*. Red. N. Bansal i I. Finocchi. T. 9294. LNCS. Springer, 2015, s. 533–544. DOI: 10.1007/978-3-662-48350-3\_45.
- [10] T. Kociumaka, T. Starikovskaya i H. W. Vildhøj. Sublinear space algorithms for the Longest Common Substring problem. [W:] *Algorithms — ESA 2014*. Red. A. S. Schulz i D. Wagner. T. 8737. LNCS. Springer, 2014, s. 605–617. DOI: 10.1007/978-3-662-44777-2\_50.
- [11] T. Kociumaka, J. Radoszewski i W. Rytter. Efficient indexes for jumbled pattern matching with constant-sized alphabet. *Algorithmica* 77.4 (2017), s. 1194–1215. DOI: 10.1007/s00453-016-0140-0.  
Wersja konferencyjna: Efficient indexes for jumbled pattern matching with constant-sized alphabet. [W:] *Algorithms — ESA 2013*. Red. H. L. Bodlaender i G. F. Italiano. T. 8125. LNCS. Springer, 2013, s. 625–636. DOI: 10.1007/978-3-642-40450-4\_53.
- [12] T. Kociumaka, S. P. Pissis i J. Radoszewski. Pattern matching and consensus problems on weighted sequences and profiles. *Theory of Computing Systems* (2018). DOI: 10.1007/s00224-018-9881-2.  
Wersja konferencyjna: Pattern matching and consensus problems on weighted sequences and profiles. [W:] *Algorithms and Computation, ISAAC 2016*. Red. S.-H. Hong. T. 64. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 46:1–46:12. DOI: 10.4230/LIPIcs.ISAAC.2016.46.
- [13] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Covering problems for partial words and for indeterminate strings. *Theoretical Computer Science* 698 (2017), s. 25–39. DOI: 10.1016/j.tcs.2017.05.026.  
Wersja konferencyjna: Covering problems for partial words and for indeterminate strings. [W:] *Algorithms and Computation, ISAAC 2014*. Red. H.-K. Ahn i C.-S. Shin. T. 8889. LNCS. Springer, 2014, s. 220–232. DOI: 10.1007/978-3-319-13075-0\_18.
- [14] T. Kociumaka, J. Pachocki, J. Radoszewski, W. Rytter i T. Waleń. Efficient counting of square substrings in a tree. *Theoretical Computer Science* 544 (2014), s. 60–73. DOI: 10.1016/j.tcs.2014.04.015.  
Wersja konferencyjna: Efficient counting of square substrings in a tree. [W:] *Algorithms and Computation, ISAAC 2012*. Red. K.-M. Chao, T.-s. Hsu i D.-T. Lee. T. 7676. LNCS. Springer, 2012, s. 207–216. DOI: 10.1007/978-3-642-35261-4\_24.
- [15] P. Gawrychowski, T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Universal reconstruction of a string. [W:] *Algorithms and Data Structures, WADS 2015*. Red. F. Dehne, J. Sack i U. Stege. T. 9214. LNCS. Springer, 2015, s. 386–397. DOI: 10.1007/978-3-319-21840-3\_32.

- [16] P. Charalampopoulos, M. Crochemore, C. S. Iliopoulos, T. Kociumaka, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Efficient enumeration of non-equivalent squares in partial words with few holes. *Journal of Combinatorial Optimization* (2018). DOI: 10.1007/s10878-018-0300-z.  
Wersja konferencyjna: Efficient enumeration of non-equivalent squares in partial words with few holes. [W:] *Computing and Combinatorics, COCOON 2017*. Red. Y. Cao i J. Chen. T. 10392. LNCS. Springer, 2017, s. 99–111. DOI: 10.1007/978-3-319-62389-4\_9.
- [17] P. Charalampopoulos, M. Crochemore, C. S. Iliopoulos, T. Kociumaka, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Linear-time algorithm for Long LCF with  $k$  mismatches. [W:] *Combinatorial Pattern Matching, CPM 2018*. Red. G. Navarro, D. Sankoff i B. Zhu. T. 105. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 23:1–23:16. DOI: 10.4230/LIPIcs.CPM.2018.23.
- [18] T. Kociumaka. Minimal suffix and rotation of a substring in optimal time. [W:] *Combinatorial Pattern Matching, CPM 2016*. Red. R. Grossi i M. Lewenstein. T. 54. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 28:1–28:12. DOI: 10.4230/LIPIcs.CPM.2016.28.
- [19] P. Gawrychowski, T. Kociumaka, W. Rytter i T. Waleń. Faster longest common extension queries in strings over general alphabets. [W:] *Combinatorial Pattern Matching, CPM 2016*. Red. R. Grossi i M. Lewenstein. T. 54. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 5:1–5:13. DOI: 10.4230/LIPIcs.CPM.2016.5.
- [20] C. Barton, T. Kociumaka, S. P. Pissis i J. Radoszewski. Efficient index for weighted sequences. [W:] *Combinatorial Pattern Matching, CPM 2016*. Red. R. Grossi i M. Lewenstein. T. 54. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, 4:1–4:13. DOI: 10.4230/LIPIcs.CPM.2016.4.
- [21] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. String powers in trees. *Algorithmica* 79.3 (2017), s. 814–834. DOI: 10.1007/s00453-016-0271-3.  
Wersja konferencyjna: String powers in trees. [W:] *Combinatorial Pattern Matching, CPM 2015*. Red. F. Cicalese, E. Porat i U. Vaccaro. T. 9133. LNCS. Springer, 2015, s. 284–294. DOI: 10.1007/978-3-319-19929-0\_24.
- [22] T. Kociumaka, J. Radoszewski i W. Rytter. Efficient ranking of Lyndon words and decoding lexicographically minimal de Bruijn sequence. *SIAM Journal on Discrete Mathematics* 30.4 (2016), s. 2027–2046. DOI: 10.1137/15M1043248.  
Wersja konferencyjna: Computing  $k$ -th Lyndon word and decoding lexicographically minimal de Bruijn sequence. [W:] *Combinatorial Pattern Matching, CPM 2014*. Red. A. S. Kulikov, S. O. Kuznetsov i P. A. Pevzner. T. 8486. LNCS. Springer, 2014, s. 202–211. DOI: 10.1007/978-3-319-07566-2\_21.
- [23] M. Babenko, P. Gawrychowski, T. Kociumaka, I. Kolesnichenko i T. Starikovskaya. Computing minimal and maximal suffixes of a substring. *Theoretical Computer Science* 638 (2016), s. 112–121. DOI: 10.1016/j.tcs.2015.08.023.  
Wersja konferencyjna: M. Babenko, P. Gawrychowski, T. Kociumaka i T. Starikovskaya. Computing minimal and maximal suffixes of a substring revisited. [W:] *Combinatorial Pattern Matching, CPM 2014*. Red. A. S. Kulikov, S. O. Kuznetsov i P. A. Pevzner. T. 8486. LNCS. Springer, 2014, s. 30–39. DOI: 10.1007/978-3-319-07566-2\_4.
- [24] T. Kociumaka, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Efficient algorithms for shortest partial seeds in words. *Theoretical Computer Science* 710 (2018), s. 139–147. DOI: 10.1016/j.tcs.2016.11.035.  
Wersja konferencyjna: Efficient algorithms for shortest partial seeds in words. [W:] *Combinatorial Pattern Matching, CPM 2014*. Red. A. S. Kulikov, S. O. Kuznetsov i P. A. Pevzner. T. 8486. LNCS. Springer, 2014, s. 192–201. DOI: 10.1007/978-3-319-07566-2\_20.
- [25] T. Kociumaka, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Fast algorithm for partial covers in words. *Algorithmica* 73.1 (2015), s. 217–233. DOI: 10.1007/s00453-014-9915-3.  
Wersja konferencyjna: Fast algorithm for partial covers in words. [W:] *Combinatorial Pattern Matching, CPM 2013*. Red. J. Fischer i P. Sanders. T. 7922. LNCS. Springer, 2013, s. 177–188. DOI: 10.1007/978-3-642-38905-4\_18.

- [26] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, M. Kubica, J. Radoszewski, W. Rytter, W. Tyczyński i T. Waleń. The maximum number of squares in a tree. [W:] *Combinatorial Pattern Matching, CPM 2012*. Red. J. Kärkkäinen i J. Stoye. T. 7354. LNCS. Springer, 2012, s. 27–40. DOI: 10.1007/978-3-642-31265-6\_3.
- [27] T. Kociumaka, J. Radoszewski, W. Rytter, J. Straszyński, T. Waleń i W. Zuba. Faster recovery of approximate periods over edit distance. [W:] *String Processing and Information Retrieval, SPIRE 2018*. Red. T. Gagie, A. Moffat i G. Navarro. T. 11147. LNCS. Springer, 2018. DOI: 10.1007/978-3-030-00478-8\_19.
- [28] M. Alzamel, P. Charalampopoulos, C. S. Iliopoulos, T. Kociumaka, S. P. Pissis, J. Radoszewski i J. Straszyński. Efficient computation of sequence mappability. [W:] *String Processing and Information Retrieval, SPIRE 2018*. Red. T. Gagie, A. Moffat i G. Navarro. T. 11147. LNCS. Springer, 2018. DOI: 10.1007/978-3-030-00478-8\_2.
- [29] G. Badkobeh, T. Gagie, S. Inenaga, T. Kociumaka, D. Kosolobov i S. J. Puglisi. On two LZ78-style grammars: compression bounds and compressed-space computation. [W:] *String Processing and Information Retrieval, SPIRE 2017*. Red. G. Fici, M. Sciortino i R. Venturini. T. 10508. LNCS. Springer, 2017, s. 51–67. DOI: 10.1007/978-3-319-67428-5\_5.
- [30] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, R. Kundu, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Near-optimal computation of runs over general alphabet via non-crossing LCE queries. [W:] *String Processing and Information Retrieval, SPIRE 2016*. Red. S. Inenaga, K. Sadakane i T. Sakai. T. 9954. LNCS. Springer, 2016, s. 22–34. DOI: 10.1007/978-3-319-46049-9\_3.
- [31] P. Gawrychowski, T. Kociumaka, W. Rytter i T. Waleń. Tight bound for the number of distinct palindromes in a tree. [W:] *String Processing and Information Retrieval, SPIRE 2015*. Red. C. S. Iliopoulos, S. J. Puglisi i E. Yilmaz. T. 9309. LNCS. Springer, 2015, s. 270–276. DOI: 10.1007/978-3-319-23826-5\_26.
- [32] H. Bannai, S. Inenaga, T. Kociumaka, A. Lefebvre, J. Radoszewski, W. Rytter, S. Sugimoto i T. Waleń. Efficient algorithms for Longest Closed Factor array. [W:] *String Processing and Information Retrieval, SPIRE 2015*. Red. C. S. Iliopoulos, S. J. Puglisi i E. Yilmaz. T. 9309. LNCS. Springer, 2015, s. 95–102. DOI: 10.1007/978-3-319-23826-5\_10.
- [33] T. Kociumaka, J. Pachocki, J. Radoszewski, W. Rytter i T. Waleń. On the string consensus problem and the Manhattan sequence consensus problem. *Theoretical Computer Science* 710 (2018), s. 126–138. DOI: 10.1016/j.tcs.2017.03.022.  
Wersja konferencyjna: On the string consensus problem and the Manhattan sequence consensus problem. [W:] *String Processing and Information Retrieval, SPIRE 2014*. Red. E. S. de Moura i M. Crochemore. T. 8799. LNCS. Springer, 2014, s. 244–255. DOI: 10.1007/978-3-319-11918-2\_24.
- [34] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, M. Kubica, A. Langiu, S. P. Pissis, J. Radoszewski, W. Rytter i T. Waleń. Order-preserving indexing. *Theoretical Computer Science* 638 (2016), s. 122–135. DOI: 10.1016/j.tcs.2015.06.050.  
Wersja konferencyjna: Order-preserving incomplete suffix trees and order-preserving indexes. [W:] *String Processing and Information Retrieval, SPIRE 2013*. Red. O. Kurland, M. Lewenstein i E. Porat. T. 8214. LNCS. Springer, 2013, s. 84–95. DOI: 10.1007/978-3-319-02432-5\_13.
- [35] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Efficient data structures for the factor periodicity problem. [W:] *String Processing and Information Retrieval, SPIRE 2012*. Red. L. Calderón-Benavides, C. N. González-Caro, E. Chávez i N. Ziviani. T. 7608. LNCS. Springer, 2012, s. 284–294. DOI: 10.1007/978-3-642-34109-0\_30.
- [36] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. On periodicity lemma for partial words. [W:] *Language and Automata Theory and Applications, LATA 2018*. Red. S. T. Klein, C. Martín-Vide i D. Shapira. T. 10792. LNCS. Springer, 2018, s. 232–244. DOI: 10.1007/978-3-319-77313-1\_18.
- [37] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Linear-time version of Holub’s algorithm for morphic imprimitivity testing. *Theoretical Computer Science* 602 (2015), s. 7–21. DOI: 10.1016/j.tcs.2015.07.055.

Wersja konferencyjna: Linear-time version of Holub’s algorithm for morphic imprimitivity testing. [W:] *Language and Automata Theory and Applications, LATA 2013*. Red. A.-H. Dediu, C. Martín-Vide i B. Truthe. T. 7810. LNCS. Springer, 2013, s. 383–394. DOI: 10.1007/978-3-642-37064-9\_34.

- [38] T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. Maximum number of distinct and nonequivalent nonstandard squares in a word. *Theoretical Computer Science* 648 (2016), s. 84–95. DOI: 10.1016/j.tcs.2016.08.010.

Wersja konferencyjna: Maximum number of distinct and nonequivalent nonstandard squares in a word. [W:] *Developments in Language Theory, DLT 2014*. Red. A. M. Shur i M. V. Volkov. T. 8633. LNCS. Springer, 2014, s. 215–226. DOI: 10.1007/978-3-319-09698-8\_19.

- [39] G. Fici, T. Kociumaka, T. Lecroq, A. Lefebvre i É. Prieur-Gaston. Fast computation of Abelian runs. *Theoretical Computer Science* 656, Part B (2016), s. 256–264. DOI: 10.1016/j.tcs.2015.12.010.

- [40] T. Flouri, C. S. Iliopoulos, T. Kociumaka, S. P. Pissis, S. J. Puglisi, W. F. Smyth i W. Tyczyński. Enhanced string covering. *Theoretical Computer Science* 506 (2013), s. 102–114. DOI: 10.1016/j.tcs.2013.08.013.

Wersja konferencyjna: New and efficient approaches to the quasiperiodic characterisation of a string. [W:] *Prague Stringology Conference 2012*. Red. J. Holub i J. Žďárek. Czech Technical University in Prague, 2012, s. 75–88.

- [41] T. Kociumaka, J. Radoszewski i B. Wiśniewski. Subquadratic-time algorithms for Abelian stringology problems. *AIMS Medical Science* 4.3 (2017), s. 332–351. DOI: 10.3934/ms.2017.3.332.

Wersja konferencyjna: Subquadratic-Time Algorithms for Abelian Stringology Problems. [W:] *Mathematical Aspects of Computer and Information Sciences, MACIS 2015*. Red. I. S. Kotsireas, S. M. Rump i C. K. Yap. T. 9582. LNCS. Springer, 2015, s. 320–334. DOI: 10.1007/978-3-319-32859-1\_27.

- [42] G. Fici, T. Kociumaka, J. Radoszewski, W. Rytter i T. Waleń. On the greedy algorithm for the Shortest Common Superstring problem with reversals. *Information Processing Letters* 116.3 (2016), s. 245–251. DOI: 10.1016/j.ipl.2015.11.015.

- [43] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, M. Kubica, J. Pachocki, J. Radoszewski, W. Rytter, W. Tyczyński i T. Waleń. A note on efficient computation of all Abelian periods in a string. *Information Processing Letters* 113.3 (2013), s. 74–77. DOI: 10.1016/j.ipl.2012.11.001.

- [44] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, M. Kubica, A. Langiu, J. Radoszewski, W. Rytter, B. Szreder i T. Waleń. A note on the longest common compatible prefix problem for partial words. *Journal of Discrete Algorithms* 34 (2015), s. 49–53. DOI: 10.1016/j.jda.2015.05.003.

## Algorytmy aproksymacyjne

- [45] A. Adamaszek, T. Kociumaka, M. Pilipczuk i M. Pilipczuk. Hardness of approximation for strip packing. *ACM Transactions on Computation Theory* 9.3 (2017), 14:1–14:7. DOI: 10.1145/3092026.

- [46] M. Cygan i T. Kociumaka. Constant factor approximation for Capacitated k-Center with Outliers. [W:] *Symposium on Theoretical Aspects of Computer Science, STACS 2014*. Red. E. W. Mayr i N. Portier. T. 25. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2014, s. 251–262. DOI: 10.4230/LIPIcs.STACS.2014.251.

- [47] M. Cygan i T. Kociumaka. Approximating upper degree-constrained partial orientations. [W:] *Approximation, Randomization, and Combinatorial Optimization APPROX/RANDOM 2015*. Red. N. Garg, K. Jansen, A. Rao i J. D. P. Rolim. T. 40. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015, s. 212–224. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2015.212.

- [48] F. Grandoni, T. Kociumaka i M. Włodarczyk. An LP-rounding  $2\sqrt{2}$ -approximation for restricted maximum acyclic subgraph. *Information Processing Letters* 115.2 (2015), s. 182–185. DOI: 10.1016/j.ipl.2014.09.008.



## Algorytmy parametryzowane i umiarkowanie wykładnicze

- [49] A. Boral, M. Cygan, T. Kociumaka i M. Pilipczuk. A fast branching algorithm for Cluster Vertex Deletion. *Theory of Computing Systems* 58.2 (2016), s. 357–376. DOI: 10.1007/s00224-015-9631-7.  
Wersja konferencyjna: A fast branching algorithm for Cluster Vertex Deletion. [W:] *Computer Science Symposium in Russia, CSR 2014*. Red. E. A. Hirsch, S. O. Kuznetsov, J.-É. Pin i N. K. Vereshchagin. T. 8476. LNCS. Springer, 2014, s. 111–124. DOI: 10.1007/978-3-319-06686-8\_9.
- [50] T. Kociumaka i M. Pilipczuk. Faster deterministic Feedback Vertex Set. *Information Processing Letters* 114.10 (2014), s. 556–560. DOI: 10.1016/j.ip1.2014.05.001.

## Algorytmy rozproszone

- [51] E. Bampas, J. Czyzowicz, L. Gąsieniec, D. Ilcinkas, R. Klasing, T. Kociumaka i D. Pająk. Linear search by a pair of distinct-speed robots. *Algorithmica* (2018). DOI: 10.1007/s00453-018-0447-0.  
Wersja konferencyjna: Linear search by a pair of distinct-speed robots. [W:] *Structural Information and Communication Complexity, SIROCCO 2016*. Red. J. Suomela. T. 9988. LNCS. Springer, 2016, s. 195–211. DOI: 10.1007/978-3-319-48314-6\_13.

## DZIAŁALNOŚĆ RECENZENCKA

---

### Członek komitetu programowego SPIRE 2017

2017

24th International Symposium on String Processing and Information Retrieval

#### *Recenzje prac konferencyjnych*

SODA, ICALP, ESA, STACS, LICS, SWAT, MFCS, ISAAC, CSR, CPM, FSTTCS, LATA, DCC, SOFSEM

#### *Recenzje prac czasopismowych*

ACM Trans. Algorithms, Theory Comput., Algorithmica, Theor. Comput. Sci., Inf. Process. Lett., J. Discrete Algorithms, Fundam. Inform., Int. J. Found. Comput. Sci., Expert Syst. Appl.