

Otoczka wypukła dla zbioru w przestrzeni dwuwymiarowej

Jakub Szymczak

Szymon Budziak

Przedstawienie problemu

Otoczka wypukła, to najmniejszy zbiór wypukły zawierający w sensie inkluzji dany zbiór punktów. Lub inaczej jest to najmniejszy wielokąt wypukły rozpięty na wierzchołkach danego zbioru punktów taki, że wszystkie punkty znajdują się albo wewnątrz niego, albo na jego krawędzi.

Wyznaczanie otoczki wypukłej dla danego zbioru punktów ma swoje liczne zastosowania w informatyce. Najpopularniejszymi są:

- Algorytm detekcji kolizji.
- Wyznaczanie najmniejszego pojemnika.
- Analiza kształtu

Algorytm Grahama

Poszczególne kroki algorytmu Grahama:

- ▶ W zbiorze S wybieramy punkt p_0 o najmniejszej współrzędnej y . Jeżeli jest kilka takich punktów, to wybieramy ten z nich, który ma najmniejszą współrzędną x .
- ▶ Sortujemy pozostałe punkty ze względu na kąt, jaki tworzy wektor (p_0, p) z dodatnim kierunkiem osi x . Jeśli kilka punktów tworzy ten sam kąt, usuwamy wszystkie z wyjątkiem najbardziej oddalonego od p_0 . Niech uzyskanym ciągiem będzie p_1, p_2, \dots, p_m .
- ▶ Do początkowo pustego stosu s wkładamy punkty p_0, p_1, p_2 . t - indeks stosu; $i \leftarrow 3$
- ▶ Iterujemy po reszcie posortowanych punktów. Jeżeli kolejny punkt znajduje się po prawej stronie odcinka utworzonego z dwóch ostatnich punktów ze stosu, lub jest z nim współliniowy, to usuwamy ostatni punkt ze stosu. W przeciwnym przypadku wstawiamy punkt na szczyt stosu i przechodzimy do następnego punktu.
- ▶ Punkty, które zostały na stosie po ukończeniu algorytmu to nasza otoczka.
- ▶ Algorytm Grahama ma złożoność $O(n \log n)$.

Algorytm Jarvisa

Poszczególne kroki algorytmu Jarvisa:

- ▶ Wyznaczenie najniżej położonego punktu spośród wszystkich. Jeśli istnieje więcej niż jeden taki punkt, to wybieramy ten o najmniejszej współrzędnej x.
- ▶ Powtarzaj następujący algorytm póki następny wykryty punkt nie jest pierwszym odcinkiem.
 - Znajdź punkt, dla którego kąt liczony przeciwnie do wskazówek zegara w odniesieniu do ostatniej krawędzi otoczki jest najmniejszy. Taki punkt jest następnym punktem należącym do otoczki.
- ▶ Punkty, które znaleźliśmy to otoczka wypukła.
- ▶ Złożoność algorytmu jest rzędu $O(kn)$ gdzie k to liczba wierzchołków otoczki.

Algorytm QuickHull

Poszczególne kroki algorytmu QuickHull:

- ▶ Znajdź dwa punkty skrajne A, B jeden o najmniejszej odciętej, a drugi o największej odciętej.
- ▶ Następnie uruchamiamy funkcję rekurencyjnego znajdowania łuku należącego do otoczki między danymi punktami należącymi do tej otoczki p, q na prawo od odcinka |p,q|. Otoczka jest sumą punktów wyniku działania funkcji rekurencyjnej dla odcinka |a,b| oraz wyniku działania funkcji rekurencyjnej dla |b,a|.
- ▶ Funkcja rekurencyjnego wyznaczania łuku należącego do otoczki między punktami p i q polega na:
 - Wyznaczaniu najbardziej oddalonego punktu na prawo od p, q, jeśli są punkty po prawej.
 - Jeśli nie ma takich punktów, to takiego łuku nie ma i zwracamy pustą tablicę.
 - W przeciwnym przypadku p, q należą do otoczki, to wyznaczony punkt skrajny r musi należeć do otoczki.
 - Skoro p,k,r należy do otoczki, to wszystkie wierzchołki wewnątrz trójkąta pkr na pewno nie należą do najmniejszej otoczki, więc je usuwamy.
 - Szukany łuk, to suma działania tej samej funkcji dla punktów p, r, punktu r, oraz wyniku funkcji dla punktów r,q w zadanej kolejności.
 - Na koniec zwracamy wyznaczony w ten sposób łuk.
- ▶ Złożoność oczekiwana $O(n \log n)$, a złożoność w najgorszym przypadku to $O(n^2)$.

Algorytm dziel i rządź

Poszczególne kroki algorytmu dziel i rządź:

- ▶ Sortujemy zbiór punktów.
- ▶ Dzielimy zbiór na mniejsze względem mediany zbioru. Powtarzamy czynność do póki wielkość dowolnego zbioru jest mniejsze niż zadane k ($k = 6$).
- ▶ Dla każdego małego zbioru wyznaczamy otoczki wypukłe innym algorytmem znajdowania otoczki wypukłej.
- ▶ Następnie łączymy otoczki wypukłe w jedno tak aby zachować złożoność algorytmu $O(n \log n)$.
- ▶ Złożoność algorytmu to $O(n \log n)$.

Algorytm przyrostowy

Poszczególne kroki algorytmu przyrostowego:

- ▶ Tworzymy z trzech pierwszych punktów otoczkę wypukłą.
- ▶ Iterujemy po pozostałych punktach wykonując następujące czynności:
 - Jeżeli punkt należy do wyznaczonej już otoczki przechodzimy do następnego punktu.
 - Jeżeli punkt nie należy do wnętrza otoczki to znajdujemy styczne do otoczki z danego punktu i aktualizujemy otoczkę o znalezione krawędzie.
- ▶ Zwracamy punkty otoczki.
- ▶ Złożoność to $O(n \log n)$.

Algorytm Chana

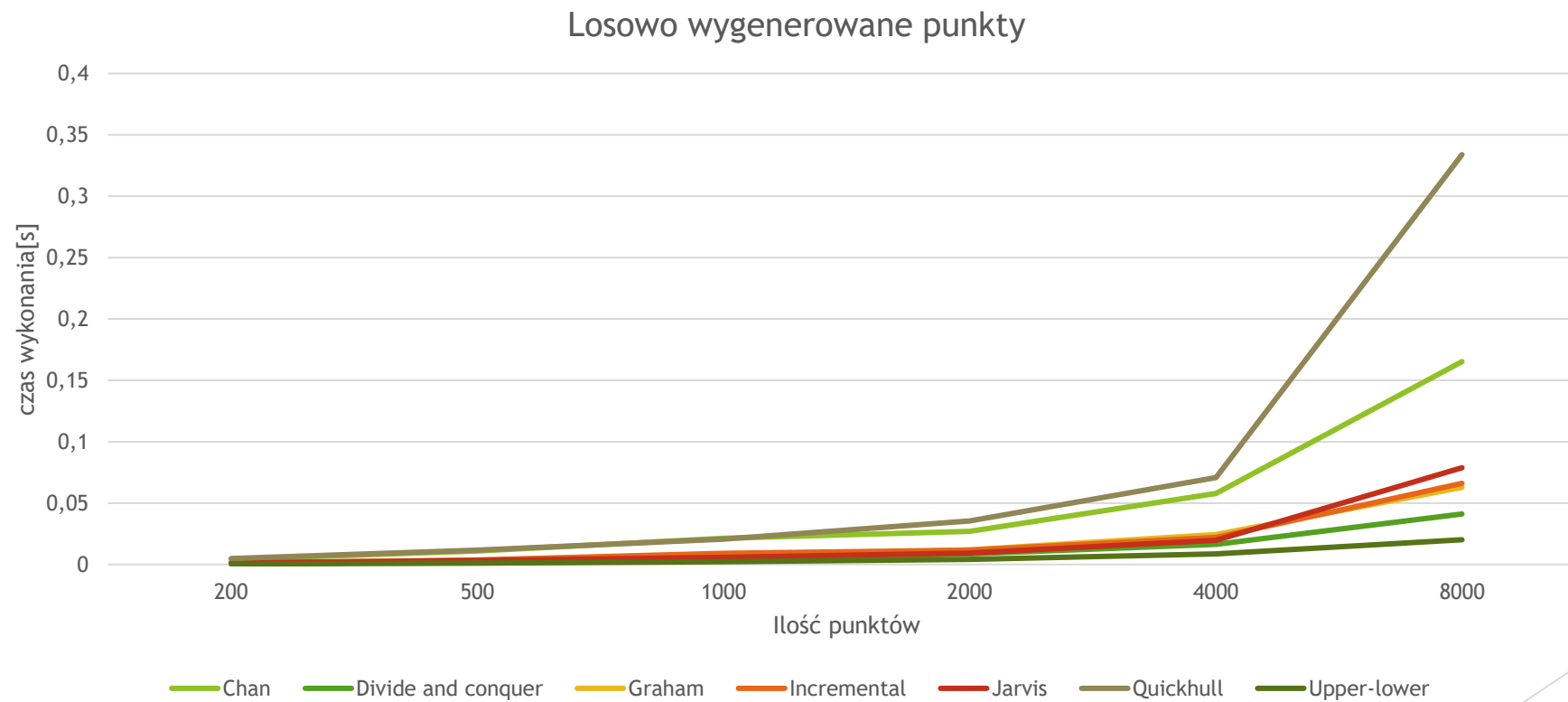
Poszczególne kroki algorytmu Chana:

- ▶ Podziel zbiór punktów na podzbiory o w miarę równych ilościach punktów w nich zawartych, z czego żaden nie zawiera więcej niż dane m .
- ▶ Wyznacz otoczki dla każdego takiego zbioru.
- ▶ Podobnie jak w algorytmie Jarvisa wybierz najniższy punkt należący do danego zbioru punktów.
- ▶ Jeżeli mamy wierzchołek należący do otoczki to możemy wyznaczyć następny:
 - Dla każdej podotoczki wyznaczamy punkt styczny do tej otoczki.
 - Spośród zbioru takich punktów oraz kolejnego punktu z podotoczki, do której dany punkt należy wybieramy taki punkt, że wszystkie pozostałe punkty znajdują się na lewo od odcinka utworzonego z wierzchołka należącego do otoczki głównej i niego.
 - Tak wybrany punkt jest kolejnym punktem otoczki głównej.
- ▶ Wyznaczamy kolejne punkty otoczki, dopóki następny punkt otoczki nie jest pierwszym wierzchołkiem otoczki. Jeżeli nie znajdziemy w ten sposób pierwszego punktu otoczki w m iteracjach, to przerywamy wykonanie algorytmu.
- ▶ Złożoność algorytmu to $O(n \log n)$.

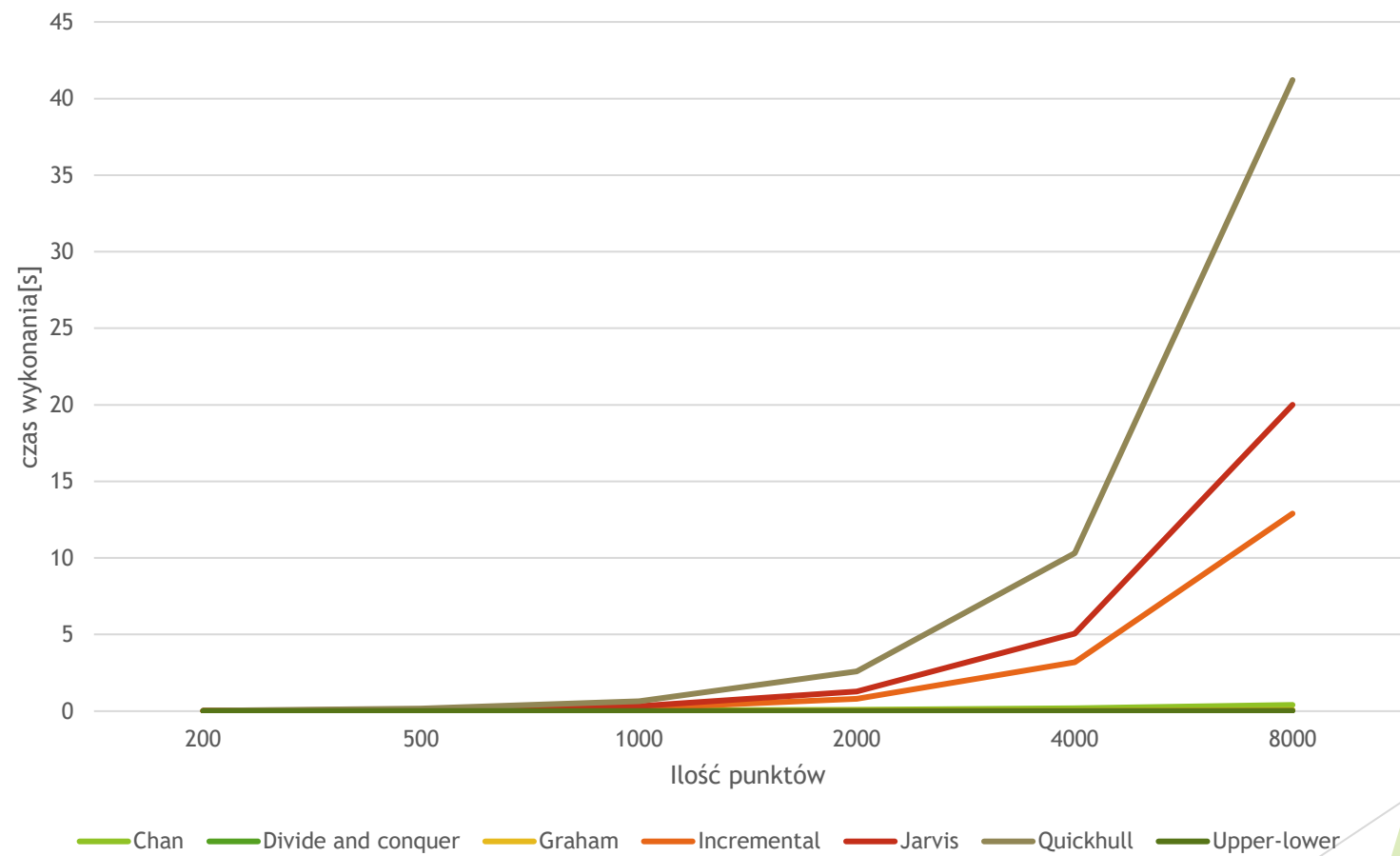
Algorytm górnej i dolnej otoczki

- ▶ Sortujemy punkty rosnąco po współrzędnych x . W przypadku remisów, porównujemy współrzędne y .
- ▶ Pierwsze dwa punkty z posortowanego zbioru wpisujemy do zbioru punktów otoczki górnej oraz dolnej.
- ▶ Iterujemy po zbiorze punktów zaczynając od trzeciego punktu:
 - Dopóki górna(dolna) otoczka ma co najmniej 2 punkty i bieżący punkt nie znajduje się po prawej(lewej) stronie odcinka skierowanego utworzonego przez ostatnie dwa punkty otoczki, to usuwamy punkt z górnej(dolnej) otoczki. W przeciwnym wypadku dodajemy punkt do otoczki górnej(dolnej).
- ▶ Odwracamy kolejność wierzchołków w otoczce dolnej.
- ▶ Łączymy zbiory punktów otoczki górnej oraz dolnej.
- ▶ Zwracamy złączony zbiór punktów otoczki.

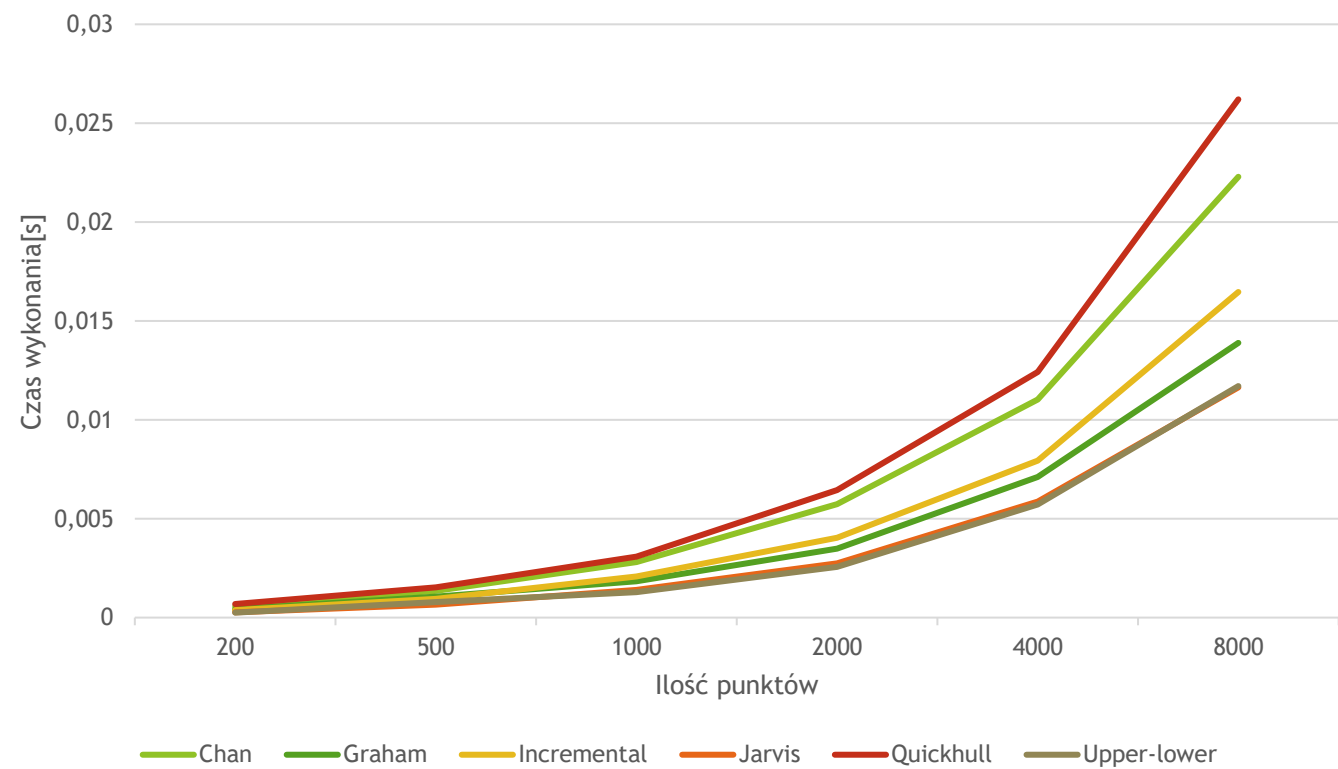
Wykresy porównania czasów działania algorytmów



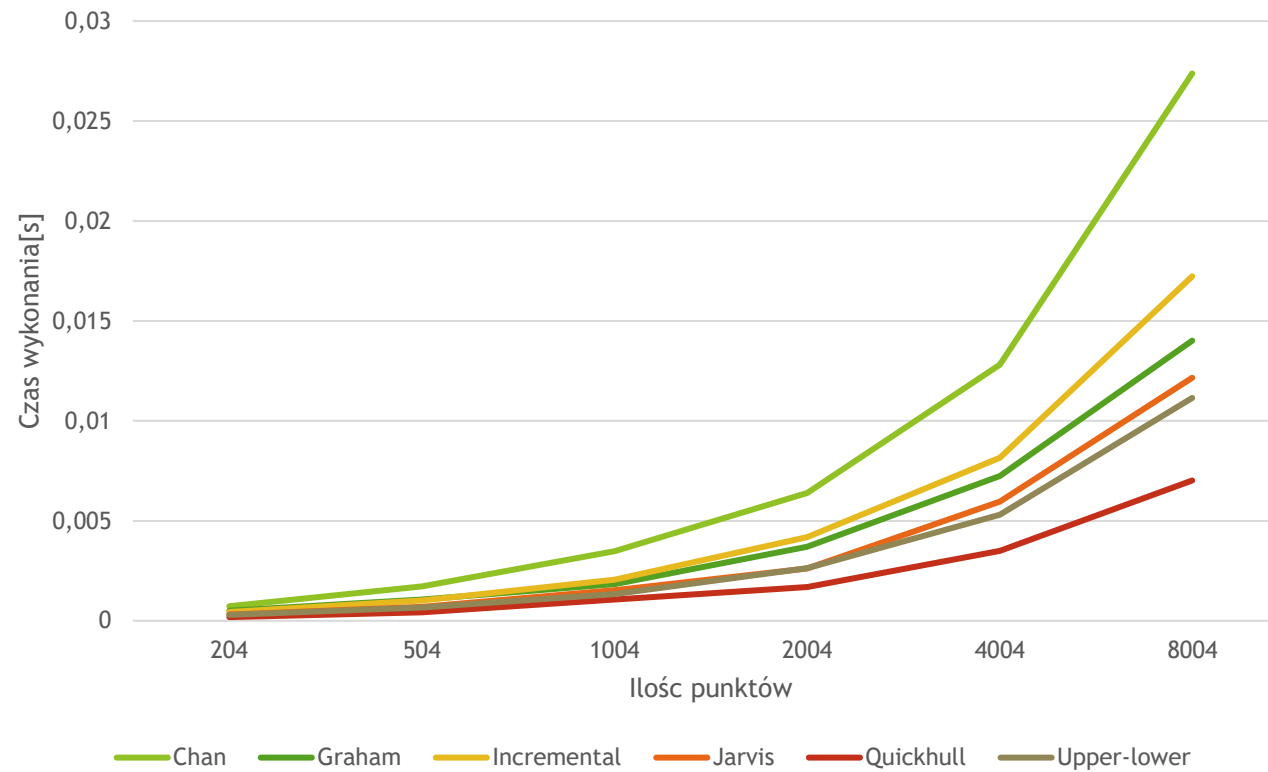
Punkty wygenerowane na okręgu



Punkty wygenerowane na prostokącie



Punkty wygenerowane na bokach kwadratu i jego przekątnych



Literatura

- ▶ Wiedza na temat wszystkich algorytmów była czerpana w większości z wykładu.