Maurycy Sosnowski (261705),

AOD Lista 03,

Sprawozdanie

Poniżej znajdują się wyniki testów dla prezentowanych przeze mnie na laboratoriach algorytmów:

1. Dijkstry z użyciem Binary Heap (biblioteka standardowa C++)
2. Algorytm Diala
3. Implementacja Radix Heap

W testach mierzony był czas działania każdego z algorytmów dla różnych rodzin instancji problemu wyszukiwania najkrótszych ścieżek. Na wykresach w niektórych momentach (niewielu) mogą pojawiać się nienaturalne „skoki”, które są najpewniej spowodowane dodatkowym obciążeniem procesora przy innych czynnościach (jak już wspomniałem jest ich zaledwie kilka ale jakież się zdarzyły.

1. **Problemy typu Long C.**
   1. **Long C (4):**

Numer of Vertexes : 1048576

Number of edges: 4063200

Max cost/weight: 256

A picture containing plot, line, screenshot, text

Description automatically generatedA picture containing text, font, screenshot, line

Description automatically generated

Wyniki Średnio (P2P):

Dijkstra BH: 945199 microseconds

Dial Algorithm: 466930 microseconds

Radix Heap: 947249 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 1976674 microseconds

Dial Algorithm: 946295 microseconds

Radix Heap: 1905083 microseconds

* 1. **Long C (5):**

Numer of Vertexes : 1048576

Number of edges: 4063200

A picture containing text, screenshot, plot, line

Description automatically generatedMax cost/weight: 1024

A picture containing text, screenshot, font, line

Description automatically generated

Wyniki Średnio (P2P):

Dijkstra BH: 1002424,365 microseconds

Dial Algorithm: 607727,1346 microseconds

Radix Heap: 1074509,096 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 1952379,4microseconds

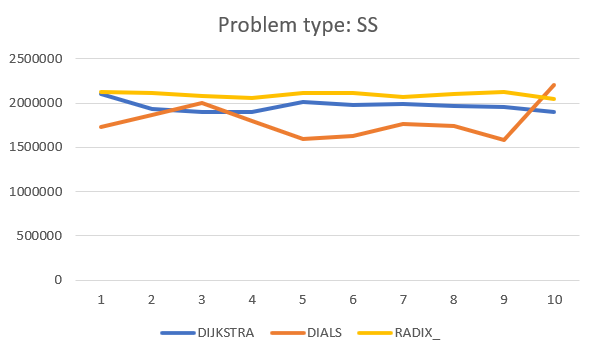
Dial Algorithm: 1121889,4 microseconds

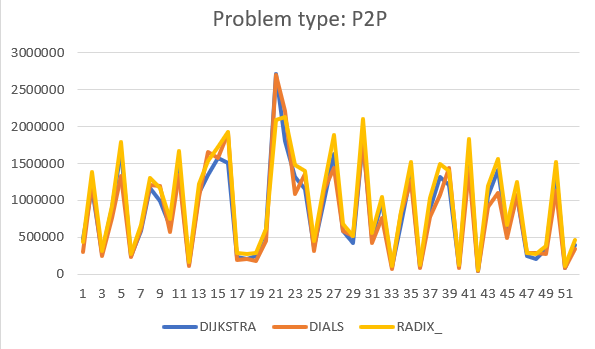
Radix Heap: 1950395 microseconds

* 1. **Long C (6):**

Numer of Vertexes : 1048576

Number of edges: 4063200

Max cost/weight: 4096



Wyniki Średnio (P2P):

Dijkstra BH: 867144,6538 microseconds

Dial Algorithm: 843220,7885 microseconds

Radix Heap: 971171,9038 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 1963646,5 microseconds

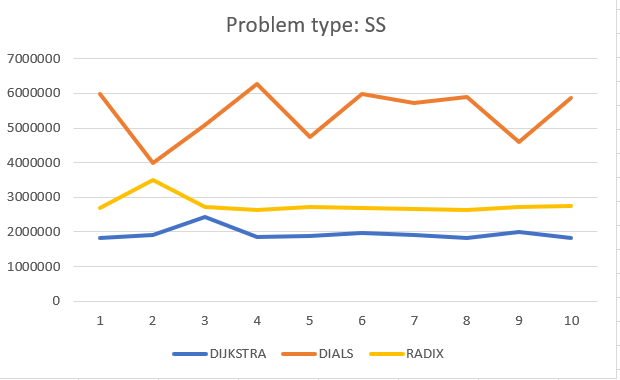
Dial Algorithm: 1791150,3microseconds

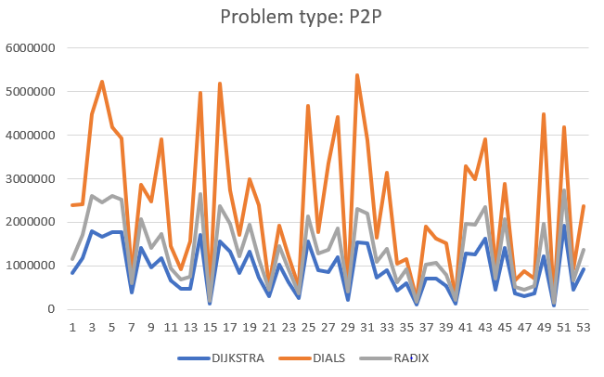
Radix Heap: 2095538,4 microseconds

* 1. **Long C (7):**

Numer of Vertexes : 1048576

Number of edges: 4063200

Max cost/weight: 16384



Wyniki Średnio (P2P):

Dijkstra BH: 917654,5769 microseconds

Dial Algorithm: 2377340,385 microseconds

Radix Heap: 1362769,519 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 1939321,6 microseconds

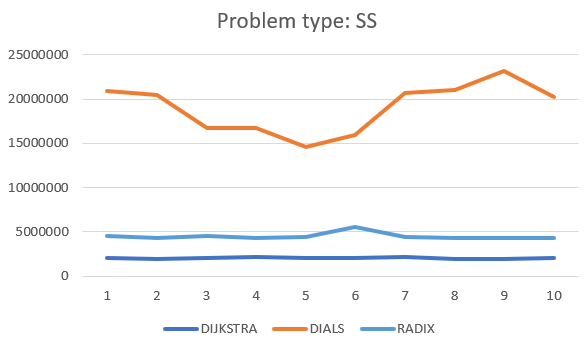
Dial Algorithm: 5415245,9 microseconds

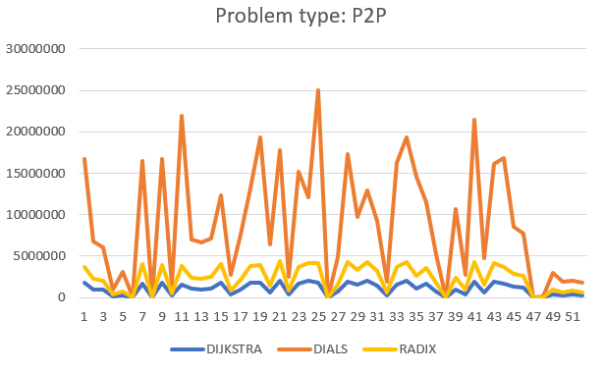
Radix Heap: 2215227,4 microseconds

* 1. **Long C (8):**

Numer of Vertexes : 1048576

Number of edges: 4063200

Max cost/weight: 65536



Wyniki Średnio (P2P):

Dijkstra BH: 1048935,308 microseconds

Dial Algorithm: 8967064,269 microseconds

Radix Heap: 2334499,017 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 2039083,4 microseconds

Dial Algorithm: 19062153,2 microseconds

Radix Heap: 4513864,18 microseconds

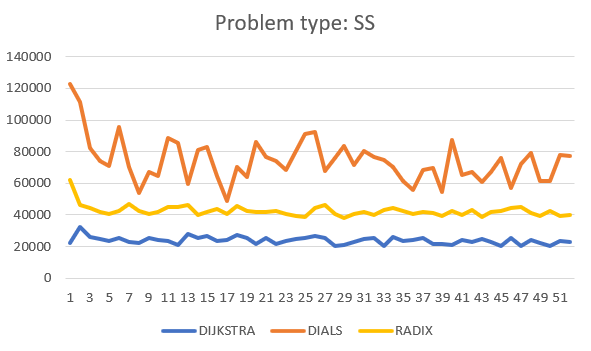
WNIOSKI / OBSERWACJE (Long C):

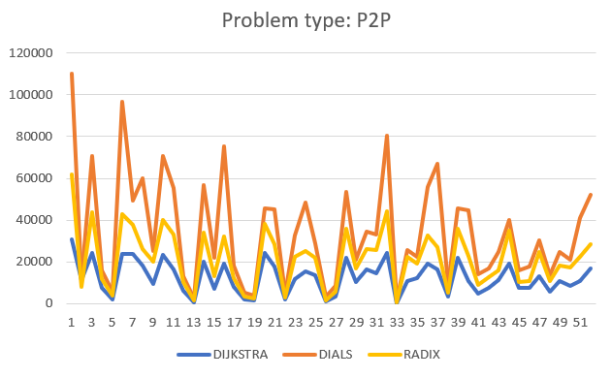
Rodzina problemów Long C ma te same ilości wierzchołków i krawędzi. Zwiększane są jedynie koszta/wagi krawędzi. Jak widać na powyższych wykresach algorytmy Diala i RadixHeap, których złożoność zależy od maksymalnego kosztu krawędzi, spowalniają wraz ze wzrostem wag. (jest to szczególnie widoczne dla algorytmu Diala zależnego liniowo, RadixHeap zależy logarytmicznie więc nie jest to aż tak widoczne).

1. **Problemy typu Long N.**
   1. **Long N (4):**

Numer of Vertexes : 16384

Number of edges: 63456

Max cost/weight: 16384



Wyniki Średnio (P2P):

Dijkstra BH: 12458,5 microseconds

Dial Algorithm: 34423,36538 microseconds

Radix Heap: 21423,49423 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 23874,86538 microseconds

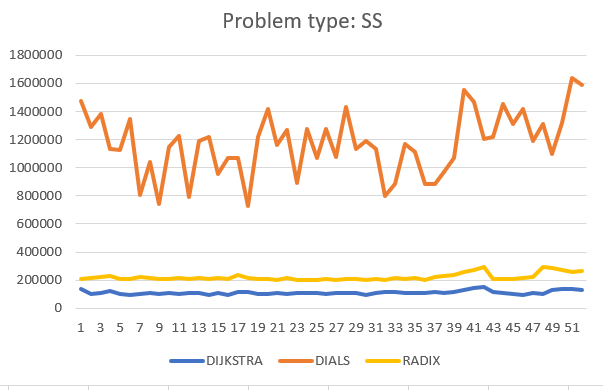
Dial Algorithm: 74032,88462 microseconds

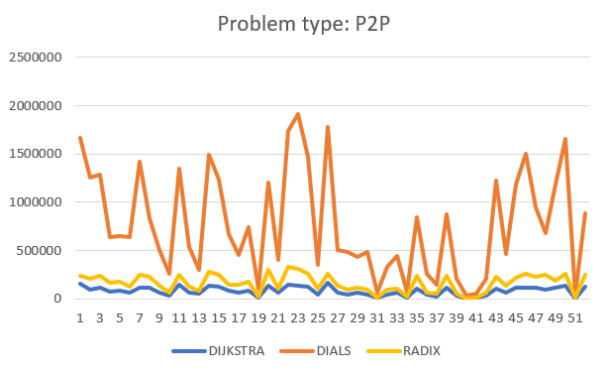
Radix Heap: 42454,25673 microseconds

* 1. **Long N (6):**

Numer of Vertexes : 65536

Number of edges: 253920

Max cost/weight: 65536



Wyniki Średnio (P2P):

Dijkstra BH: 80666,67 microseconds

Dial Algorithm: 770876,7692 microseconds

Radix Heap: 161756,9764 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 110370,4038 microseconds

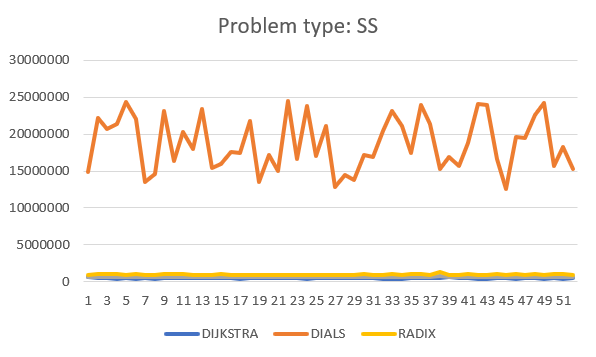
Dial Algorithm: 1169222,269 microseconds

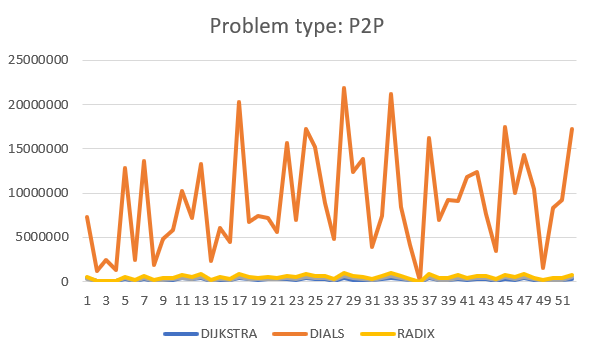
Radix Heap: 221002,975 microseconds

* 1. **Long N (8):**

Numer of Vertexes : 262144

Number of edges: 1015776

Max cost/weight: 262144



Wyniki Średnio (P2P):

Dijkstra BH: 341699 microseconds

Dial Algorithm: 17265337 microseconds

Radix Heap: 750980,475 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 456907,3846 microseconds

Dial Algorithm: 18711339,94 microseconds

Radix Heap: 976055,6332 microseconds

WNIOSKI / OBSERWACJE (Long N):

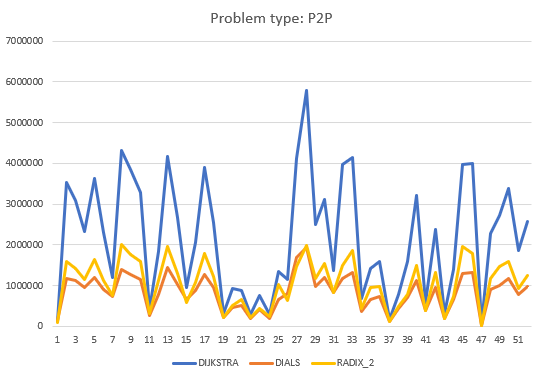
Rodzina problemów Long N ma raczej małe ilości krawędzi. Maksymalny koszt/waga krawędzi ( C ) i ilość wierzchołków maję te samą wartość . Jak widać na powyższych wykresach algorytm Diala ma zdecydowanie większą złożoność niż DijkstraBH i RadixHeap (Teoretycznie V = C, zatem dla Diala O(E + VC) = O(E+V^2), dla RadixHeap O(E + V log(VC)) = O(E + V(log(V^2)). Najlepiej radzi tu sobie Dijkstra z Binary heapem;

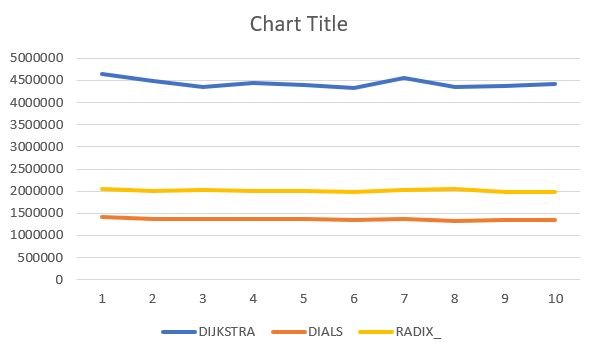
1. **Problemy typu Random4 C.**
   1. **Random4 C (4):**

Numer of Vertexes : 1048576

Number of edges: 4194304

Max cost/weight: 256





Wyniki Średnio (P2P):

Dijkstra BH: 2154790,75 microseconds

Dial Algorithm: 839511,8846 microseconds

Radix Heap: 1242117,808 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 4435082,5 microseconds

Dial Algorithm: 1365292,9 microseconds

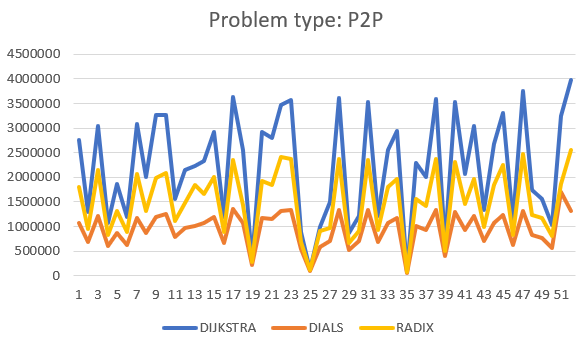
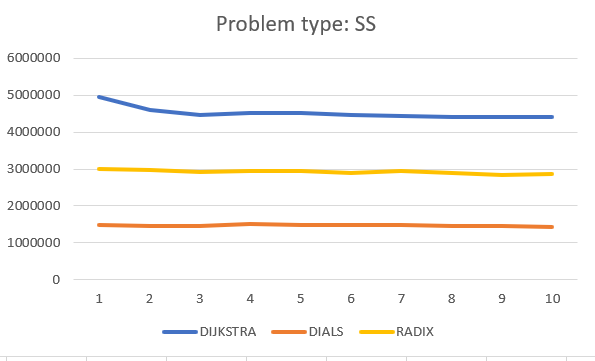
Radix Heap: 2109423,9 microseconds

1. **Random4 C (6):**

Numer of Vertexes : 1048576

Number of edges: 4194304

Max cost/weight: 4096



Wyniki Średnio (P2P):

Dijkstra BH: 2209325,115 microseconds

Dial Algorithm: 940283,8462 microseconds

Radix Heap: 1504844,285 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 4519975,7 microseconds

Dial Algorithm: 1469202,3 microseconds

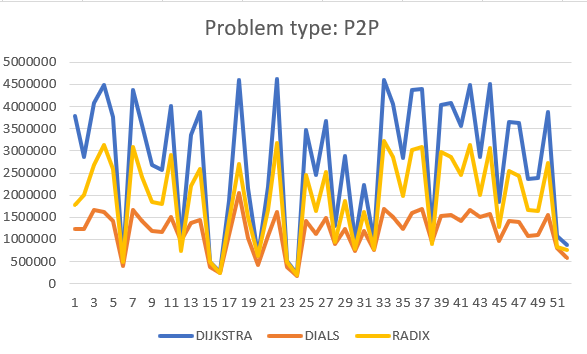
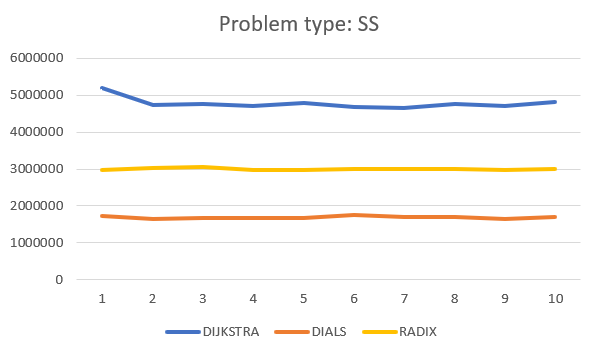
Radix Heap: 2924325,36 microseconds

1. **Random4 C (8):**

Numer of Vertexes : 1048576

Number of edges: 4194304

Max cost/weight: 65536



Wyniki Średnio (P2P):

Dijkstra BH: 2788182,327 microseconds

Dial Algorithm: 1200284,442 microseconds

Radix Heap: 1940444,372 microseconds

Wyniki Średnio (SS):

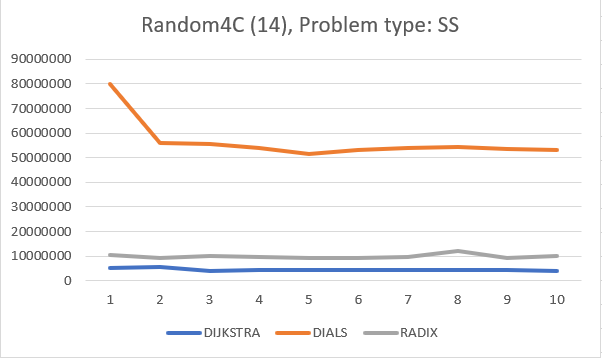
Dijkstra BH: 4776586,8 microseconds

Dial Algorithm: 1689130,3 microseconds

Radix Heap: 2999024,84 microseconds

WNIOSKI / OBSERWACJE (Random4 C):

Rodzina problemów Random4 C, Podobnie jak w Long C, ma te same ilości wierzchołków i krawędzi. Zwiększane są jedynie koszta/wagi krawędzi. Na powyższych wykresach algorytmy Diala i RadixHeap, których złożoność zależy od maksymalnego kosztu krawędzi, spowalniają wraz ze wzrostem wag. Dla „małych”(praktycznie to nawet sporych) C radzą sobie jednak lepiej od dijkstry z binary heapem. Spowolnienie (w zależności od Max C) jednak następuje tutaj dużo wolniej niż w Long C, co wynika zapewne ze sposobu w jaki te grafy są generowane. Spowolnienie wyraźnie widać dopiero np. dla instancji = 14, które (tylko w wersji SS) zamieszczam dodatkowo poniżej.

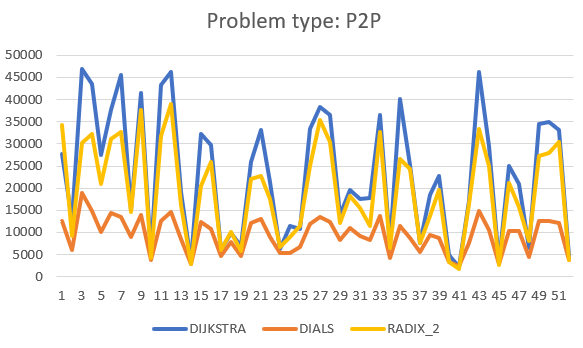


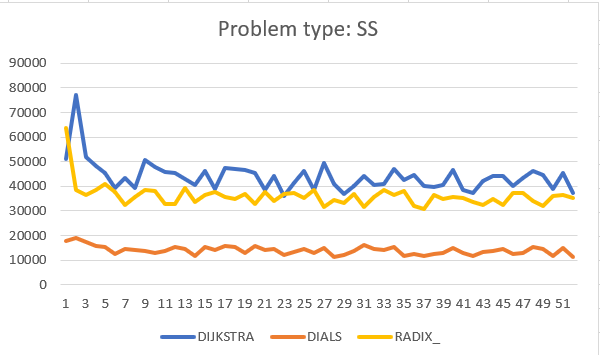
1. **Problemy typu Random4 N.**
   1. **Random4 N (4):**

Numer of Vertexes : 16384

Number of edges: 65536

Max cost/weight: 16384





Wyniki Średnio (P2P):

Dijkstra BH: 23189,25 microseconds

Dial Algorithm: 9468,326923 microseconds

Radix Heap: 19038,76923 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 43869,98077 microseconds

Dial Algorithm: 13957,32692 microseconds

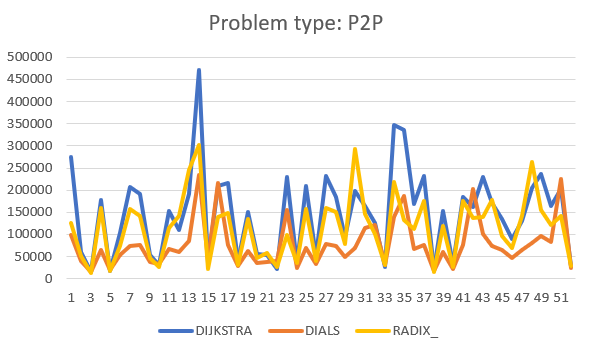
Radix Heap: 36070,48077 microseconds

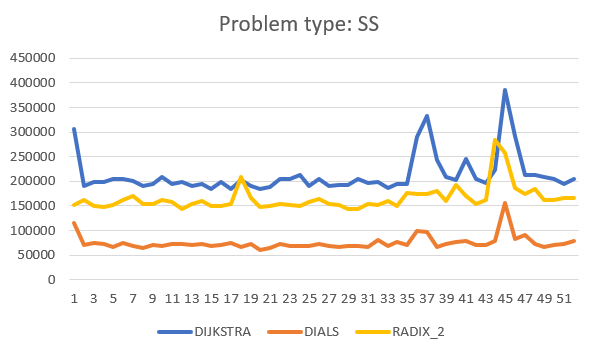
* 1. **Random4 N (6):**

Numer of Vertexes : 65536

Number of edges: 262144

Max cost/weight: 65536





Wyniki Średnio (P2P):

Dijkstra BH: 146368,4423 microseconds

Dial Algorithm: 77344,05769 microseconds

Radix Heap: 114652,5192 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 212381,6346 microseconds

Dial Algorithm: 75010,96154 microseconds

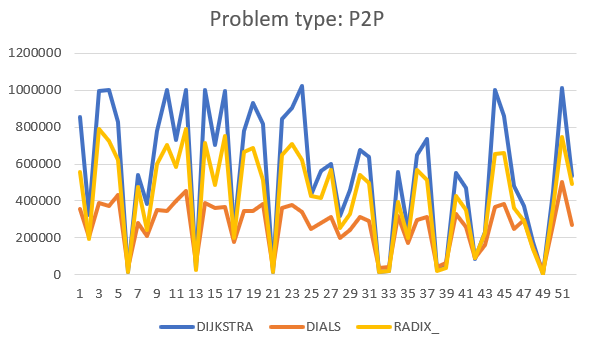
Radix Heap: 164989,1346 microseconds

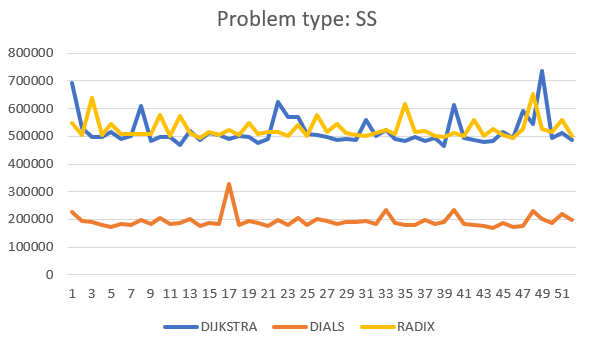
* 1. **Random4 N (8):**

Numer of Vertexes : 262144

Number of edges: 1048576

Max cost/weight: 262144





Wyniki Średnio (P2P):

Dijkstra BH: 556696,4231 microseconds

Dial Algorithm: 267056,8269 microseconds

Radix Heap: 421793,5192 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 1003268,154 microseconds

Dial Algorithm: 409293,0769 microseconds

Radix Heap: 760438,3077 microseconds

WNIOSKI / OBSERWACJE (Random4 N):

Rodzina problemów Random4n jest teoretycznie podobna do omawianej wyżej, rodziny Long n : maksymalny koszt C = V i liczba krawędzi E nie jest zbyt duża. Tutaj jednak, w przeciwieństwie do wspomnianej wcześniej Long n, wyniki nieco się różnią – algorytm Diala okazuje się tu sprawować najlepiej nawet dla późniejszych, większych instancji co mija się z wynikami dla Long n. Możliwe, że różnice te wynikają z różnicy struktur tych grafów (były one generowane w różny sposób). Jeżeli chodzi o radixHeap i algorytm Dijkstry z binary heapem to radzą sobie one podobnie (radixHeap nieco lepiej).

1. **Problemy typu Road Graph(USA)d/t.**
   1. **USA NY (d)**

Numer of Vertexes : 264346

Number of edges: 733846

Max cost/weight: 92366

A picture containing text, font, screenshot, line

Description automatically generated

A picture containing text, screenshot, font, plot

Description automatically generated

Wyniki Średnio (P2P):

Dijkstra BH: 250887,5385 microseconds

Dial Algorithm: 93171,30769 microseconds

Radix Heap: 231446,4038 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 517677,8654 microseconds

Dial Algorithm: 193190,1538 microseconds

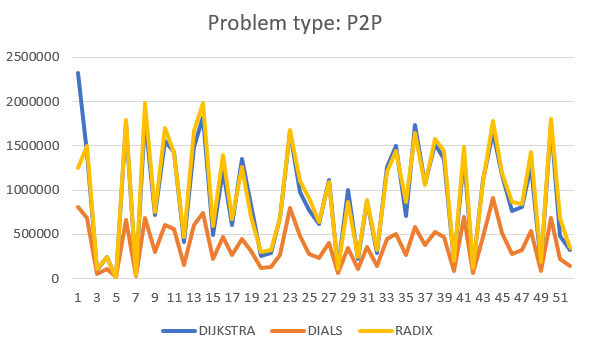
Radix Heap: 525680,5962 microseconds

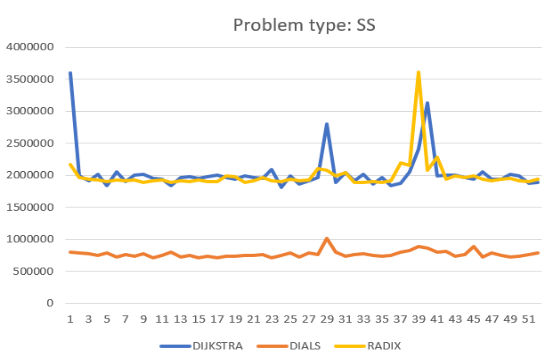
* 1. **USA FLA (d)**

Numer of Vertexes : 1070376

Number of edges: 2712798

Max cost/weight: 214013





Wyniki Średnio (P2P):

Dijkstra BH: 939907,1538 microseconds

Dial Algorithm: 378376,8654 microseconds

Radix Heap: 961914,2019 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 2035678,962 microseconds

Dial Algorithm: 769883,8269 microseconds

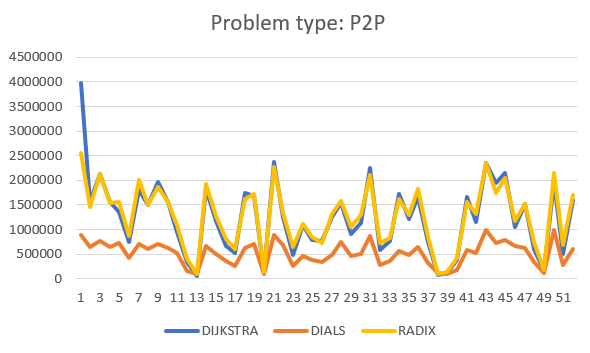
Radix Heap: 1992084,067 microseconds

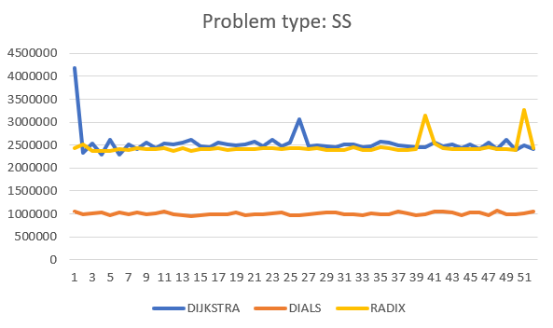
* 1. **USA NW (t)**

Numer of Vertexes : 1207945

Number of edges: 2840208

Max cost/weight: 265941





Wyniki Średnio (P2P):

Dijkstra BH: 1251060,692 microseconds

Dial Algorithm: 520886,9423 microseconds

Radix Heap: 1267937,769 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 2535215,558 microseconds

Dial Algorithm: 1005913,231 microseconds

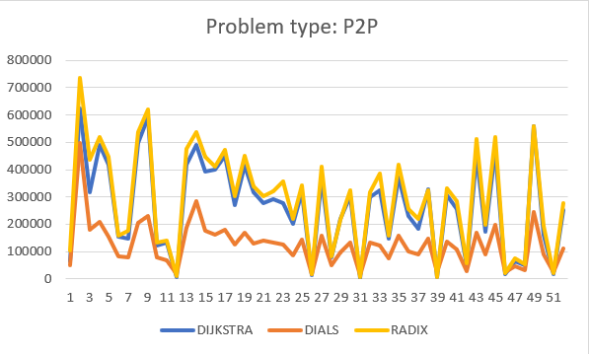
Radix Heap: 2446538,692 microseconds

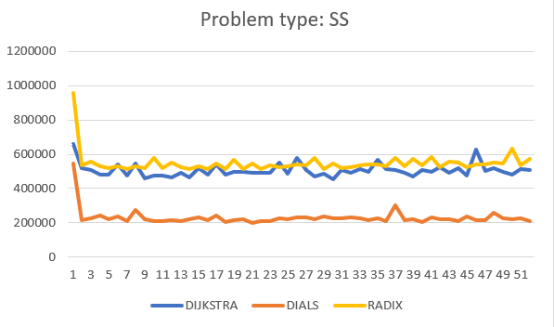
* 1. **USA BAY (t)**

Numer of Vertexes : 321270

Number of edges: 800172

Max cost/weight: 235763





Wyniki Średnio (P2P):

Dijkstra BH: 264877,9038 microseconds

Dial Algorithm: 124923,6346 microseconds

Radix Heap: 292941,9231 microseconds

Wyniki Średnio (SS):

Dijkstra BH: 505922,1346 microseconds

Dial Algorithm: 229978,3846 microseconds

Radix Heap: 543952,6462 microseconds

WNIOSKI / OBSERWACJE (USA d/t):

Rodzina problemów USA reprezentuje realne sieci połączeń drogowych. Jak widać w praktyce najlepiej radzi tu sobie algorytm Diala, co wynika zapewne z faktu, że maksymalne koszta: C połączeń nie są bardzo duże (zazwyczaj ok. 200 000). Jeżeli chodzi o radix heapa to radzi on sobie podobnie do Dijkstry z binary heapem.

WNIOSKI / OBSERWACJE OGÓLNE

Testując różne algorytmy wyszukiwania najkrótszych ścieżek, możemy zauważyć, że złożoność teoretyczna nie zawsze ma przełożenie w praktyce. Algorytm Diala, mimo nie tak dobrej złożoności teoretycznej (ogólnie raczej gorszej niż np. radixHeap), w praktyce radził sobie najlepiej w większości testowanych problemów włączając w to praktyczne przypadki jak rzeczywista sieć dróg USA. Jego wady wychodziły dopiero dla specyficznych instancji problemu (jak np. te z rodziny Long C i Long N).

Ogólnie algorytmy radzą sobie lepiej lub gorzej w zależności od rodzaju/rodziny problemów, dlatego w praktyce warto zapoznać się z rodzajem sieci/grafu z jakim mamy do czynienia i dobrać do niej odpowiedni algorytm.