

Lidia Sobońska 310903
Arkadiusz Kojtek 310746

AISDI

Kopce

Sprawozdanie

Oświadczamy że poniższa praca, będąca podstawą do ocenienia wyników trzecich laboratoriów z AISDI została wykonana przez nas samodzielnie.

Biblioteki wykorzystane w projekcie

Do stworzenia naszego projektu korzystaliśmy z następujących modułów:

1. **pyplot** z **Matplotlib**, który został wykorzystany do narysowania wykresów przebiegów czasowych.
2. **sys** służący do przyjęcia informacji o pliku źródłowym czy pliku docelowym z wynikami.
3. **timeit** wykorzystany do zmierzenia czasu tworzenia każdego z kopców.
4. **random**, który posłużył do wygenerowania zestawu losowych liczb.
5. **igraph**, który posłużył do wygenerowania wizualizacji gotowego kopca.

Algorytm tworzenia kopca

W pliku **heap.py** znajdują się funkcje odpowiedzialne za tworzenie kopców n-arnych.

create_heap() - tworzenie kopca; jako argumenty przyjmuje dane, z który ma być utworzony kopiec oraz arność wyjściowego kopca

insert_node() - wstawianie elementu do kopca

draw_heap() - wizualizacja kopca oraz zapisanie jej do pliku

Podział prac

Tworzenie kopca zostało napisane przez Arkadiusza Kojtek, wizualizacja kopca oraz pomiary czasowe przez Lidię Sobońską.

Sposób uruchomienia

Plikiem obsługującym każdy z projektów jest main.py, uruchomienie projektu polega na wywołaniu pliku main.py z poziomu terminala z dopisanymi dwoma opcjonalnymi argumentami - main.py --outfile [nazwa_pliku] --infile [nazwa_pliku] reprezentującymi odpowiednio plik, do którego zostaną zapisane przebiegi czasowe oraz plik z opcjonalnymi danymi źródłowymi. Jeżeli plik z danymi źródłowymi nie zostanie podany, tworzenie kopców

będzie pracować na losowo wygenerowanym zestawie liczb. Jeżeli natomiast nie zostanie podany plik wyjściowy, to domyślnie przebiegi czasowe zostaną zapisane do pliku **times.tsv**.

Przebieg działania programu

Projekt najpierw próbuje wczytać dane, z których mają być tworzone z wybranego przez użytkownika pliku, po czym przechodzi do mierzenia czasu tworzenia kopców oraz dodawania do nich elementów. Czas mierzony jest z użyciem funkcji `timeit()`. Informacje na temat wielkości kolekcji z danymi oraz czasu wykonania konkretnych funkcji są na bieżąco zapisywane do tablic, które następnie wykorzystywane są do utworzenia wykresu z wykorzystaniem obiektu `pyplot`. Następnie tworzone są przykładowe kopce 2,3,4-arne na mniejszej liczbie danych w celu zaprezentowania na wizualizacji (wizualizacja przy dużej liczbie elementów staje się nieczytelna).

Pliki wynikowe

Jako plik wynikowy zostaną utworzone pliki:

create_2ary.png - przebieg czasowy tworzenia kopca 2-arnego

create_3ary.png - przebieg czasowy tworzenia kopca 3-arnego

create_4ary.png - przebieg czasowy tworzenia kopca 4-arnego

create_heaps.png - porównawczy przebieg czasowy tworzenia wszystkich kopców

heap_2ary.png - wizualizacja skończonego kopca 2-arnego

heap_3ary.png - wizualizacja skończonego kopca 3-arnego

heap_4ary.png - wizualizacja skończonego kopca 4-arnego

Analiza danych

Pliki z wynikami czasowymi oraz wizualizacją kopców w związku z dużą ilością pomiarów i przez to słabą czytelnością zostaną zamieszczone jako oddzielne pliki .tsv oraz .png

Ze względu na niski czas tworzenia się kopców i przez to niemiernodajne wyniki, każda próba powtarzana była 200 razy, żeby dać bardziej wiarygodne zobrazowanie przebiegów czasowych.

Można zauważyć, że przebiegi czasowe dla wykonanego przez nas algorytmu są liniowe, jednak wyraźnie zauważalna jest różnica między poszczególnymi kopcami względem ich arności. Im kopiec mniej arny tym dłuższy czas jego tworzenia. Wiąże się to ze znacznie szybciej rosnącą wysokością takiego kopca i przez to większą ilością operacji jakie algorytm musi wykonać zanim upewni się, że kolejny element tablicy nie wymaga umieszczenia go w wyższym węźle kopca.





