ZMP - Lista 3

Marcin Michalski, WMAT PWr

Marzec 2024

Zadania tradycyjnie należy rozwiązać zgodnie ze specyfikacją i udostępnić prowadzącym laboratoria.

Deadline: ???

Ćwiczenie 1. Zaimplementuj samodzielnie funkcję, która w zadanej tablicy liczb znajduje `max` i `min`. Zrób to na 2 sposoby - pętlą i rekurencyjnie.

Ćwiczenie 2. Zaimplementuj następującą funkcję aproksymującą e

$$\sum_{k=0}^{n} \frac{1}{k!}.$$

Zrób to na 2 sposoby - pętlą i rekurencyjnie. Postaraj się, by Twoje funkcje nie liczyły dla każdego $k \leqslant n$ liczby $k!$.

Ćwiczenie 3. Zaimplementuj algorytmy sortowania

1. bąbelkowego,
2. przez wstawianie,
3. quick sort,
4. merge sort.

Zastanów się nad możliwymi usprawnieniami w porównaniu do podstawowych wersji omówionych na wykładzie. Oszacuj potrzebną liczbę istotnych porównań i przypisań w zależności od wielkości danych.

Rozwiązania zadań umieść w `lista_3/z_i`, gdzie `i` to numer zadania, w swoim repozytorium. Pliki nazwij `nr_indeksu_zic.cpp`, gdzie `c`, to odpowiedni podpunkt, o ile zadanie taki ma (jeśli nie ma, to `c` należy pominąć). Przykład: `123456_z1a.cpp`.

Zadanie 1. Gra w zapałki polega na tym, że dwoje graczy na przemian zabiera z ustalonej puli zapałek $1, 2$ lub 3 zapałki. Przegrywa gracz, który jest zmuszony wziąć ostatnią zapałkę.

(a) Zaimplementuj podstawowy wariant tej gry, w którym program zapyta o liczbę zapałek, a następnie będzie czekał na ruchy graczy aż do ich (zapałek, nie graczy) wyczerpania.

(b) ⋆ Zaimplementuj opcję gry z graczem komputerowym, który, o ile to możliwe, stosuje optymalną strategię. Opisz tę strategię.

Zadanie 2. Zaimplementuj wariant *merge sorta*, który pomija początkowe dzielenie tablicy na pół i zaczyna od razu "od dołu", tzn. na początku scala singletony tablicy, potem powstałe w ten sposób pary, czwórki, etc. Program powinien oczekiwać ze standardowego wejścia wielkość tablicy, a potem tyle liczb całkowitych, ile ta wielkość wynosi, a na końcu wyświetlić posortowaną tablicę oddzielając kolejne liczby spacjami.