zadanie nr 4 termin realizacji: 21 marzec 2005

C++

sortowanie przybliżone (klasa z konstruktorem kopiującym i destruktorem) studia dzienne

Zdefiniuj klasę `Tablica`, która będzie służyć do trzymania w pamięci tablicy liczb całkowitych o zadanym rozmiarze. Rozmiar tablicy określ poprzez parametr w konstruktorze. Konstruktor powinien wypełnić komórki tablicy kolejnymi liczbami całkowitymi rozpoczynając od 0. Dodatkowo zdefiniuj konstruktor kopiujący i destruktor w tej klasie.

```
class Tablica
{
protcted:
    int *tab ;
    int rozm ;
public:
    Tablica (int n) ;
    Tablica (const Tablica &t) ;
    ~Tablica () ;
public:
    void losowaPermutacja () ;
    long long liczbaInwersji () ;
// ...
} ;
```

Klasa `Tablica` powinna również posiadać metodę `losowaPermutacja` do losowego wymieszania elementów w tablicy (postaraj się, aby każda permutacja pojawiała się z jednakowym prawdopodobieństwem) oraz metodę `liczbaInwersji`, która zlicza ile jest różnych par nieuporządkowanych elementów (trzeba rozpatrzyć każdą parę; w sumie jest ich $\frac{n(n-1)}{2}$ w tablicy n-elementowej).

Następnie napisz program, który będzie symulował proces sortowania przybliżonego. Zaczynamy od n-elementowej tablicy z losowo wymieszanymi elementami, dla której obliczany liczbę inwersji. Następnie przepuszczamy tą tablicę przez warstwę zawierającą $\frac{n}{2}$ losowo rozstawionych komparatorów (każdy element bierze udział w porównaniu) i znowu badamy liczbę inwersji. Proces ten powtarzamy, aż liczba inwersji spadnie poniżej n.

Komparator działa w ten sposób, że porządkuje parę liczb: jeśli para liczb na wybranych pozycjach i oraz j (zakładamy, że $i < j$) występuje w porządku rosnącym, to zostawiamy te liczby na swoich miejscach, a w przeciwnym przypadku dokonujemy ich zamiany elementów. Warstwa, to taki układ komparatorów, że żadne dwa nie operują na tej samej pozycji (coś jak skojarzenie w grafie). Postaraj się coraz bardziej ograniczać długości komparatorów (czyli maksymalną odległość pomiędzy pozycjami i oraz j) w kolejnych fazach działania programu, na przykład:

$$|j - i| < \log n \cdot \sqrt{\text{liczba insersji w poprzedniej fazie}}$$

W wyniku program powinien wypisać na standardowe wyjście liczbę inwesji w tablicy po jej przejściu przez każdą warstwę. Uruchom program dla różnych wartości n rozmiaru tablicy. Wartość tą przekaż do programu poprzez parametr wywołania. Spróbuj dla $n = 1000, 10000, 100000, 1000000$.

Co się tyczy generatora liczb pseudolosowych, to w bibliotece `<cstdlib>` są zdefiniowane dwie użyteczne funkcje: `srand` i `rand`. Pierwsza z nich służy do zainicjalizowania generatora, a druga do wylosowania nieujemnej liczby typu `int`. Oto prosty przykład ich wykorzystania:

```
# include <iostream>
# include <iomanip>
# include <cstdlib>

using namespace std ;

int main ()
{
    srand((unsigned)time(0)) ;
    for (int i=0; i<4; i++)
    {
        int x = rand()%1000 ; // x to losowa liczba z zakresu 0..999
        cout<<setw(4)<<x<<endl ; // wypisanie liczby x na 4 pozycjach
    }
}
```

Uniwersytet Wrocławski
Instytut Informatyki
ul. Przesmyckiego 20
PL-51-151 Wrocław

Paweł Rzechonek