

Egzamin z Zaawansowanego Programowania Komputerowego

10 czerwca 2024

1. (15 pkt.) Napisać funkcję

```
void a(int* t, int n);
```

która zamienia wszystkie sekwencje kolejnych jednakowych elementów na 0. Np. $a(\{1,1,2,3,3,3\},6)$ zamienia tablicę na $\{0,0,2,0,0,0\}$, $a(\{7,1,1,7,1,7\},6)$ zamienia tablicę na $\{7,0,0,7,1,7\}$, $a(\{7,1,0,1,0\},5)$ nie dokonuje żadnych zmian. Oszacować złożoność napisanej funkcji.

2. (15 pkt.) Napisać funkcję

```
int f(int a, int b);
```

która zwraca liczbę napisów o następujących własnościach:

- napis składa się z dokładnie a liter 'A', dokładnie b liter 'B' i żadnych innych znaków,
- w każdym prefiksie napisu występuje więcej lub tyle samo liter 'A' co liter 'B'.

Np. $f(2,2)=2$ (napisy „AABB”, „ABAB”), $f(3,2)=5$ (napisy „AAABB”, „AABAB”, „AABBA”, „ABAAB”, „ABABA”).

3. (10 pkt.) Dana jest funkcja

```
void g(int* t, int n, int k) {  
    cout << t[k];  
    t[k]--;  
    if(t[k]>0) g(t,n, (k+t[k])%n);  
}
```

Napisać równoważną funkcję, która nie używa rekurencji.

4. (10 pkt.) Dana jest struktura

```
class ElListy { public: int wartosc; ElListy* nast; };
```

której będziemy używać do tworzenia list jednokierunkowych.

Napisać funkcję

```
bool sufiks(ElListy* a, ElListy* b);
```

która zwraca true wtedy i tylko wtedy, gdy lista a jest sufiksem listy b, tj. ostatnie elementy listy b są takie same jak cała lista a. Np. lista $1 \rightarrow 2$ jest sufiksem $3 \rightarrow 1 \rightarrow 1 \rightarrow 2$, ale nie jest sufiksem $1 \rightarrow 2 \rightarrow 1$; lista pusta jest sufiksem każdej listy.

5. (15 pkt.) Dana jest struktura

```
class Wezel{ public: int wartosc; Wezel* lewy; Wezel* prawy; };
```

której będziemy używać do tworzenia drzew.

Napisać funkcję

```
int sumaLisci(Wezel* d);
```

która zwraca sumę wartości zapisanych w liściach: węzłach, które nie mają poddrzew.

6. (15 pkt.) Stworzyć klasę Znajomi, której obiekty reprezentują sieć znajomości pomiędzy osobami. Osoby identyfikowane są przez liczby $0,1,\dots,n-1$. Znajomość jest symetryczna, tj. jeśli osoba a zna osobę b to osoba b zna osobę a.

```
class Znajomi {  
public:  
    Znajomi(int n);  
    void poznaj(int a, int b);  
    bool zna(int a, int b);  
    int max();  
    vector<int> znajomi(int a);  
    int wspolniZnajomi(int a, int b);  
};
```

Opis metod publicznych, które należy napisać:

`Znajomi(int n);` – tworzy grupę osób o numerach $0,\dots,n-1$, żadne dwie się nie znają;

`void poznaj(int a, int b);` – poznaje osoby a i b;

`bool zna(int a, int b);` – zwraca true jeśli osoby a i b się znają;

`int max();` – zwraca numer osoby, która ma najwięcej znajomych (jeśli jest remis, którąkolwiek);

`vector<int> znajomi(int a);` – zwraca wektor zawierający numery znajomych osoby a;

`int wspolniZnajomi(int a, int b);` – zwraca liczbę wspólnych znajomych osób a i b.