Pravila

Kolokvij se piše 120 minuta. Ukupno je moguće ostvariti 100 bodova. Pismeni dio predajete uživo. Rješenja zadataka spremite u datoteke zad_*i*.cpp, arhivirajte ih u zip datoteku *prezime_ime*.zip i pošaljite na e-mail.

Zadatak 1. Koristeči Master teorem, dajte asimptotske ocjene na sljedeća vremena izvršavanja.

1. $T(n) = 3T(n/2) + n^2$,
2. $T(n) = 4T(n/2) + n^2$,
3. $T(n) = T(n/2) + 2^n$,
4. $T(n) = 2^n T(n/2) + n^n$,
5. $T(n) = 16T(n/4) + n$,
6. $T(n) = 2T(n/2) + n \log n$,
7. $T(n) = 2T(n/2) + n/\log n$,
8. $T(n) = 2T(n/4) + n^{0.51}$,
9. $T(n) = 0.5T(n/2) + 1/n$,
10. $T(n) = 16T(n/4) + n!$,
11. $T(n) = \sqrt{2}T(n/2) + \log n$,
12. $T(n) = 3T(n/3) + \sqrt{n}$,
13. $T(n) = 4T(n/2) + cn$,
14. $T(n) = 3T(n/4) + n \log n$,
15. $T(n) = 3T(n/3) + n/2$,
16. $T(n) = 6T(n/3) + n^2 \log n$,
17. $T(n) = 4T(n/2) + n/\log n$,
18. $T(n) = 64T(n/8) - n^2 \log n$,
19. $T(n) = 7T(n/3) + n^2$,
20. $T(n) = 4T(n/2) + \log n$,
21. $T(n) = T(n/2) + n(2 - \cos n)$,
22. $T(2n) = T(n/2) + n$.

Zadatak 2. Dajte gornje i doljnje asimptotske granice za sljedeća vremena izvršavanja.

1. $T(n) = T(n-2) + n$,
2. $T(n) = T(n-1) + T(n-3) + \log n$,
3. $T(n) = T(n-1) + 2T(n-2) + 1$,
4. $T(n) = T(n/3) + T(2n/3) + n$,
5. $T(n) = T(n/4) + T(n/7) + \sqrt{n}$,
6. $T(n) = T(n-1) + T(1) + n$.

Zadatak 3. Tribonacci je niz prirodnih brojeva dan sljedećom rekurzivnom formulom.

$$\begin{aligned}
F(0) &= 0, \\
F(1) &= 0, \\
F(2) &= 1, \\
F(n) &= F(n-1) + F(n-2) + F(n-3), \ (\forall n > 2)\,.
\end{aligned}$$

Vrijeme izvršavanja algoritma koji na naivan način računa n-ti Tribonaccijev broj dan je rekurzijom

$$T(n) = T(n-1) + T(n-2) + T(n-3) + c.$$

Metodom stabla rekurzije dajte gornju i doljnju asimptotsku granicu na vrijeme izvršavanja tog algoritma. Indukcijom dokažite da je vrijeme izvršavanja naivnog algoritma je $O(2^n)$.
Kao i kod Fibonacci, za Tribonacci postoji matrica $A \in \mathbb{R}^{3\times 3}$ takva da vrijedi

$$A \begin{bmatrix} F(n-2) \\ F(n-1) \\ F(n) \end{bmatrix} = \begin{bmatrix} F(n-1) \\ F(n) \\ F(n+1) \end{bmatrix}.$$

Dajte matricu A i implementirajte funkciju koja u vremenu $\Theta(\log n)$ računa n-ti Tribonaccijev broj.

Zadatak 4. Dano je malo izmjenjeno brzo potenciranje

$$\begin{aligned}
x^0 &= 1, \\
x^n &= \begin{cases} \left(x^{n/3}\right)\left(x^{n/3}\right)\left(x^{n/3}\right), & n\%3 == 0, \\ x\left(x^{n-1}\right), & n\%3 == 1, \\ x^2\left(x^{n-2}\right), & n\%3 == 2. \end{cases}
\end{aligned}$$

Kratko opišite algoritam koji će pomoću te formule računa n-tu potenciju danog broja x i metodom stabla rekurzije dajte asimptotsko vrijeme izvršavanja tog algoritma. Vrijeme izvršavanja mora biti brže od linearnog.

Zadatak 5. Kratko opišite algoritam koji u vremenu $\Theta(n \log n)$ računa broj inverzija u polju A duljine n. Inverzija je par pozicija (i, j) za koji vrijedi $i < j$, $A[i] > A[j]$.

Zadatak 6. Implementirajte funkciju

```
    int int_log10(int n);
```

koja u vremenu $\Theta(\log n)$ računa $\lfloor \log_{10}(n) \rfloor$.

Zadatak 7. Implementirjate funkciju

```
    int int_sqrt(int n);
```

koja u vremenu $\Theta(\log n)$ računa $\lfloor \sqrt{n} \rfloor$.

Zadatak 8. Dan je vektor brojev A duljine n. Implementirajte funkciju

```
    int min_abss(const vector<int> &A);
```

koja u vremenu $O(n \log n)$ vrati broj x za koji je suma

$$\sum_{i=0}^{n-1} |A[i] - x|$$

najmanja.

Zadatak 9. Implementirajte funkciju

```
    int even_divs(int n);
```

koja u vremenu $O(\sqrt{n})$ vrača broj prirodnih brojeva manjih ili jednakih n koji imaju paran broj dijelitelja. (Može se izračunati i u $O(\log n)$).

Zadatak 10. Implementirajte strukturu BST (binary search tree) koja podržava sljedeće operacije.

```
class BST {
    ...
public:
    void insert(int); \\ ubacuje element u BST
    bool search(int); \\ pojavljuje li se element u BST
    void erase(int); \\ izbaci element iz BST
    int min(); \\ koji je najmanji element BST
    int max(); \\ koji je najveci element BST
    int lt(int); \\ koji je prvi manji od danog elementa
    int gt(int); \\ koji je prvi veci od danog elementa
};
```

Za svaku od metoda dajte worst-case i best-case asimptotske ocjene vremena izvršavanja.

Zadatak 11. Implementirajte minimalno orijentiranu binarnu hrpu Heap pomoću vektora koja podržava sljedeće operacije.

```
class Heap {
    vector<int> H;
    ...
public:
  int top(); \\ vrati gornji/najmanji element hrpe
  void pop(); \\ makne gornji/najmanji element hrpe
  void push(int); \\ ubaci broj u hrpu
};
```

Svaka od implementiranih metoda mora imati vrijeme izvršavanja $O(\log n)$.

Zadatak 12. Kada Matko ima slobodnog vremena ide u knjižnicu čitati knjige. Danas ima t minuta slobodnog vremena i zato je Matko uzeo n knjiga iz knjižnice i za svaku procijenio koliko mu treba vremena da ju pročita. Za knjigu i mu treba $A[i]$ minuta.

Matko je odlučio izabrati jednu knjigu i od nje će početi čitati. Ako je izabrao knjigu i, onda će pročitati knjige $i, i+1, i+2$ itd. sve dok mu ne istekne slobodno vrijeme ili dok ne pročita zadnju knjigu. Ukoliko mu istekne slobodno vrijeme u pola čitanja knjige, onda tu knjigu neće brojati kao pročitanu.

Pomozite Matku da izračuna koliko najviše knjiga može pročitati ako izabere optimalnu početnu knjigu.

Dan je vektor A veličine $1 \leq n \leq 2 \cdot 10^5$ i slobodno vrijeme $1 \leq t \leq 10^9$. Vrijeme čitanja knjige će biti prirodan broj $1 \leq A[i] \leq 10^4$. Svoje rješenje implementirajte kao funkciju

```
    int books(const vector<int> &A, int t);
```

koja u vremenu $\Theta(n)$ računa najveći broj knjiga koje Matko može pročitati. U danom algoritmu, koliko je amortizirano vrijeme izvršavanja za da jednu knjigu isprobamo kao početnu knjigu (amortizirano vrijeme po knjizi)?