Užduotys

1. Funkcijų augimo asimptotiniai žymėjimai ir jų apibrėžimai. Pateikti pavyzdžių. Naudodamiesi apibrėžimus patikrinkite ar $T(n) = \Theta(n)$ yra $T(n) = T(k) + T(n-k-1) + d$ lygties sprendinys, d – konstanta.
2. Rekurentinių sąryšių sprendimo būdai (aprašyti idėją). Suformuluoti Pagrindinę teoremą. Pateikite pavyzdį lygties, kurios $f(n) = n^3$, o sprendinys $T(n) = \Theta(n^3 \log_4 n)$
3. Rikiavimo piramide algoritmo idėja. Kokia duomenų struktūra – „piramidė“? Kaip priklauso piramidės dydis ir aukštis nuo rikiuojamų duomenų kiekio? Kaip atliekamas piramidės sutvarkymas (pateikite algoritmą ir sudėtingumą įrodykite).
4. Optimalūs rikiavimo algoritmai naudojantys palyginimus. Įrodyti, kad rikiavimo piramide ir suliejimo būdu yra asimptotiškai optimalūs algoritmai.
5. Rikiavimas su įterpimu idėja. Įrodyti rikiavimas su įterpimu algoritmo korektiškumą ir jo sudėtingumą kai duomenys įvedami palankiausiu ir nepalankiausiu atveju.
6.
   a) Suprastinti funkcionalą: $O\left(\sqrt{n}\ln n + \sqrt[4]{n^3}\right)$;
   b) Ar galima žemiau pateiktą lygtį išspęsti taikant pagrindinę teoremą? Atsakymą pagrįskite ir jei galima išspęskite.
   $$T(n) = 3T\left(\frac{n}{3}\right) + n\log_2 n$$
7.
   a) Suprastinti funkcionalą: $O\left(n^2\log_2 n + \left(\sum_{i=\frac{n}{2}}^{n} i^2\right)\right)$.
   b) Išspręsti rekurentinę lygtį: $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{3n}{4}\right) + cn$
8.
   a) Palyginti $f(n) = n^2\log_3 n$ ir $g(n) = \frac{n}{\log_2^2 n}$ funkcijų augimą.
   b) Suraskite sumą: $\sum_{k=m}^{n} k^3$.
9.
   a) Palyginti funkcijas: $f(n) = \frac{n}{\ln(n+1)}$, $g(n) = \frac{n+5}{\ln\ n}$.
   b) Išspręsti rekurentinę lygtį $T(m) = 2T\left(\frac{m}{3}\right) + \log_3 m^3$. Lygtį išspęskite taikydami pagrindinę teoremą.
10.
    a) Įvertinkite sumą: $\sum_{i=a}^{b} \frac{c}{i}$
    čia a, b, c teigiami sveiki skaičiai, be to $3a < b$.
    b) Išspręsti lygtį, taikydami sprendinių medžio metodą: $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + cn^2$.
11. Išspręsti lygtį, taikydami sprendinių medžio metodą: $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{3}\right) + cn^2$.
12. Taikydami pagrindinę teoremą išspęskite lygtį: $T(n) = 10T\left(\frac{n}{3}\right) + e^{2\ln n}\ln n$
13. Išspręsti lygtį, taikydami sprendinių medžio metodą: $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$.
14. Taikydami pagrindinę teoremą išspęskite lygtį: $T(n) = 2T\left(\frac{2n}{3}\right) + e^{2\ln n}$
15. Išspręsti lygtį, taikydami sprendinių medžio metodą: $T(n) = T\left(\frac{3n}{4}\right) + T\left(\frac{2n}{3}\right) + cn$.
16. Suraskite P1 procedūros apatinį ir viršutinį asimptotinius įvertinimus.

```
int P1(int length, int k) {
    int sum = 0;
    for (int i = 0; i < length; i++)
```

```
    {
        sum += P2(i, k);
    }
    return sum;
}
int P2(int x, int a)
{
    if (x < a) return a;
    return P2(x / 2, a) + 1;
}
```

17. Suraskite A1 procedūros apatinį ir viršutinį asimptotinius įvertinimus.

```
int A1(int length, int k)
{
    int sum = 0;
    for (int i = 0; i < length; i++)
    {
        sum += A2(i, k);
    }
    return sum;
}
int A2(int x, int a)
{
    if (x < a) return a;
    return A2(x - 2, a) + 1;
}
```

18. Suraskite K1 procedūros apatinį ir viršutinį asimptotinius įvertinimus.

```
int K1( int k)
{
    int sum = 0;
    for (int i = 0; i < k; i++)
    {
        sum += K2(i, k);
    }
    return sum;
}
int K2(int x, int a)
```

```
    {
        if (x < a) return a;
        return K2(x - 5, a) + 1;
    }
```

19. Suraskite S1 procedūros apatinį ir viršutinį asimptotinius įvertinimus.

```
    int S1(int length, int k)
    {
        int sum = 0;
        for (int i = length; i <= k; i++)
        {
            sum += S2(i, k);
        }
        return sum;
    }
    int S2(int x, int a)
    {
        if (x < a) return a;
        int sum = 0;
        for (int i = 0; i < x; i++)
        {
            sum += 1;
        }
        return S2(x / 2, a) + S2(x / 2, a) + sum;
}
```

20. Suraskite D1 procedūros apatinį ir viršutinį asimptotinius įvertinimus.

```
    int D1(int length, int k)
    {
        int sum = 0;
        for (int i = length; i <= k; i++)
        {
            sum += D2(i, k);
        }
        return sum;
    }
    int D2(int x, int a)
    {
```

```
    if (x < a) return a;
    int sum = 0;
    for (int i = 0; i < x; i++)
    {
        sum += 1;
    }
    return D2(x / 5, a) + D2(x / 5, a) + sum;
}
```