PROBLEM REZANJA ŠIPKE

empirijska i teorijska analiza algoritma, rekurzivno i dinamičko programiranje

Maja Pavičić

- Šipka duljine n
- Besplatni rezovi
- Svakoj duljini šipke pridružena vrijednost
- Određujemo način rezanja šipke koji maksimizira dobit

OPIS PROBLEMA

```
CUT_ROD(p,n)
    if n == 0
        return 0

    q = -inf
    for i = 1 to n
        q = max{q,p[i] + CUT_ROD(p, n-i)}
    return q
```

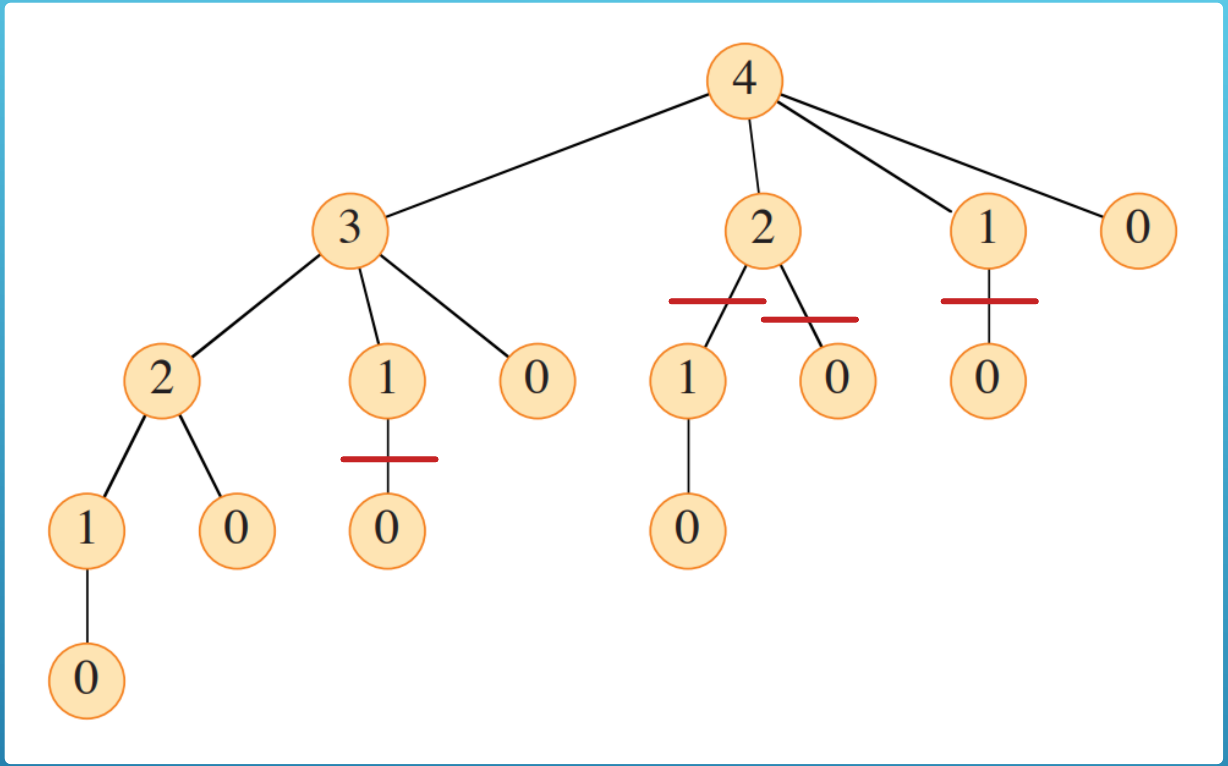
- $T(n) \coloneqq$ *broj poziva funkcije CUT_ROD za zadanu početnu duljinu štapa n*
- $T(n) = 1 + \sum_{j=0}^{n-1} T(j)$
- $T(n) = 2 \cdot T(n-1)$
- $T(n) = 2^n \cdot T(0) = 2^n$
- $\Longrightarrow\ T(n) \in \theta(2^n)$

REKURZIVNO RJEŠENJE I TEORIJSKA ANALIZA SLOŽENOSTI

```
MEMOIZED_CUT_ROD(p, n)
    let r[0:n] be a new array
    for i = 0 to n
        r[i] = -inf
    return MEMOIZED_CUT_ROD_AUX(p, n, r)
```

```
MEMOIZED_CUT_ROD_AUX(p, n, r)
    if r[n] >= 0
        return r[n]
    if n == 0
        q = 0
    else
        q = -inf
        for i = 1 to n
            q = max{q, p[i], MEMOIZED_CUT_ROD_AUX(p, n-i, r)}
        r[n] = q
        return q
```

MEMOIZIRANO REKURZIVNO RJEŠENJE



- *Broj poziva funkcije MEMOIZED_CUT_ROD_AUX*
- $T(n) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$
- $T(n) \in \theta(n^2)$

TEORIJSKA ANALIZA SLOŽENOSTI MEMOIZIRANOG REKURZIVNOG ALGORITMA

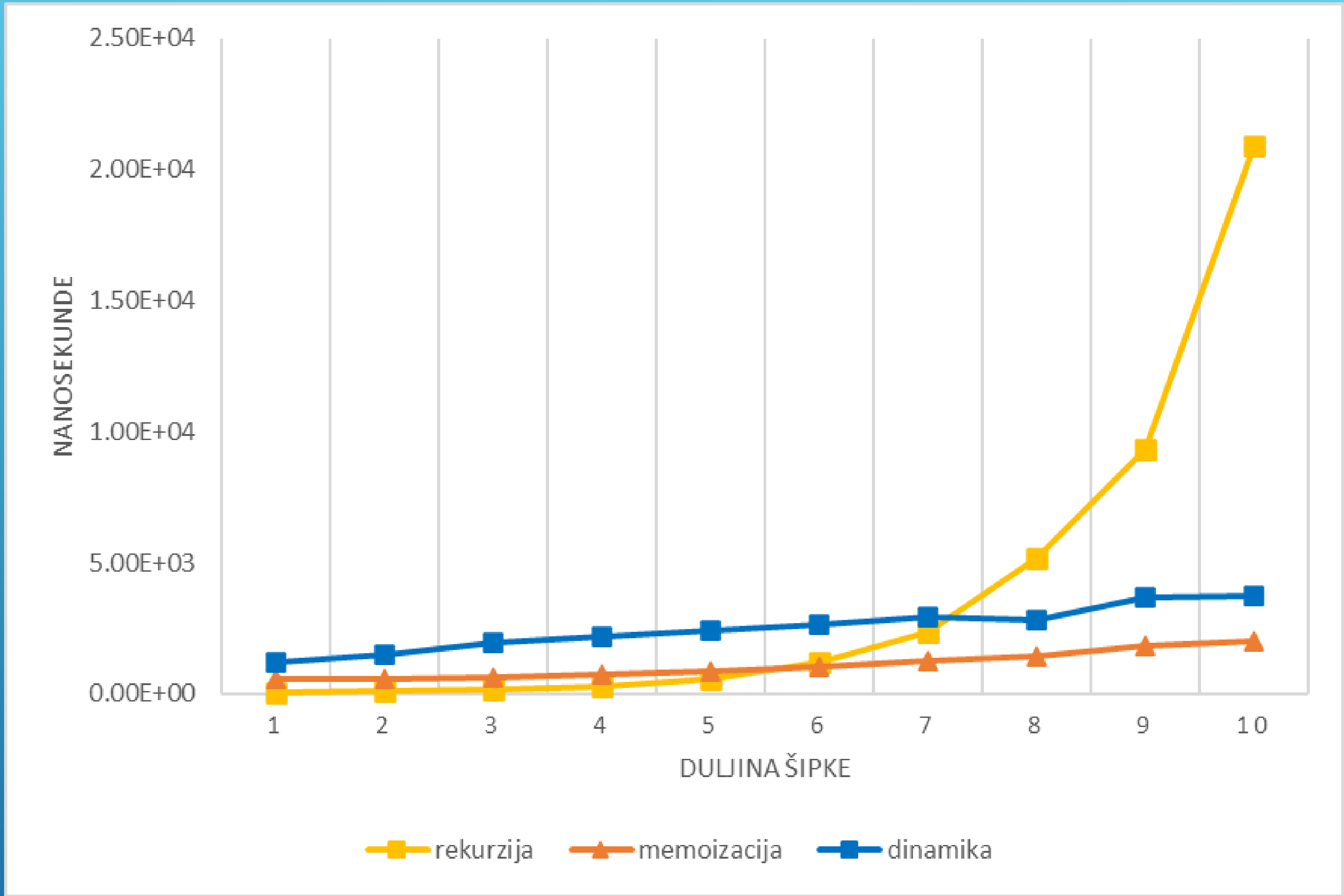
```
BOTTOM_UP_CUT_ROD(p, n)
    let r[0:n] be a new array
    r[0] = 0

    for j = 1 to n
        q = -inf
        for i = 1 to j
            q = max{q, p[i] + r[j-i]}
        r[j] = q

    return r[n]
```

- Broj operacija određivanja maksimuma
- $T(n) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$
- $T(n) \in \theta(n^2)$
- Manje konstante nego u memoiziranom rješenju – izbjegnuti skupi funkcijski pozivi

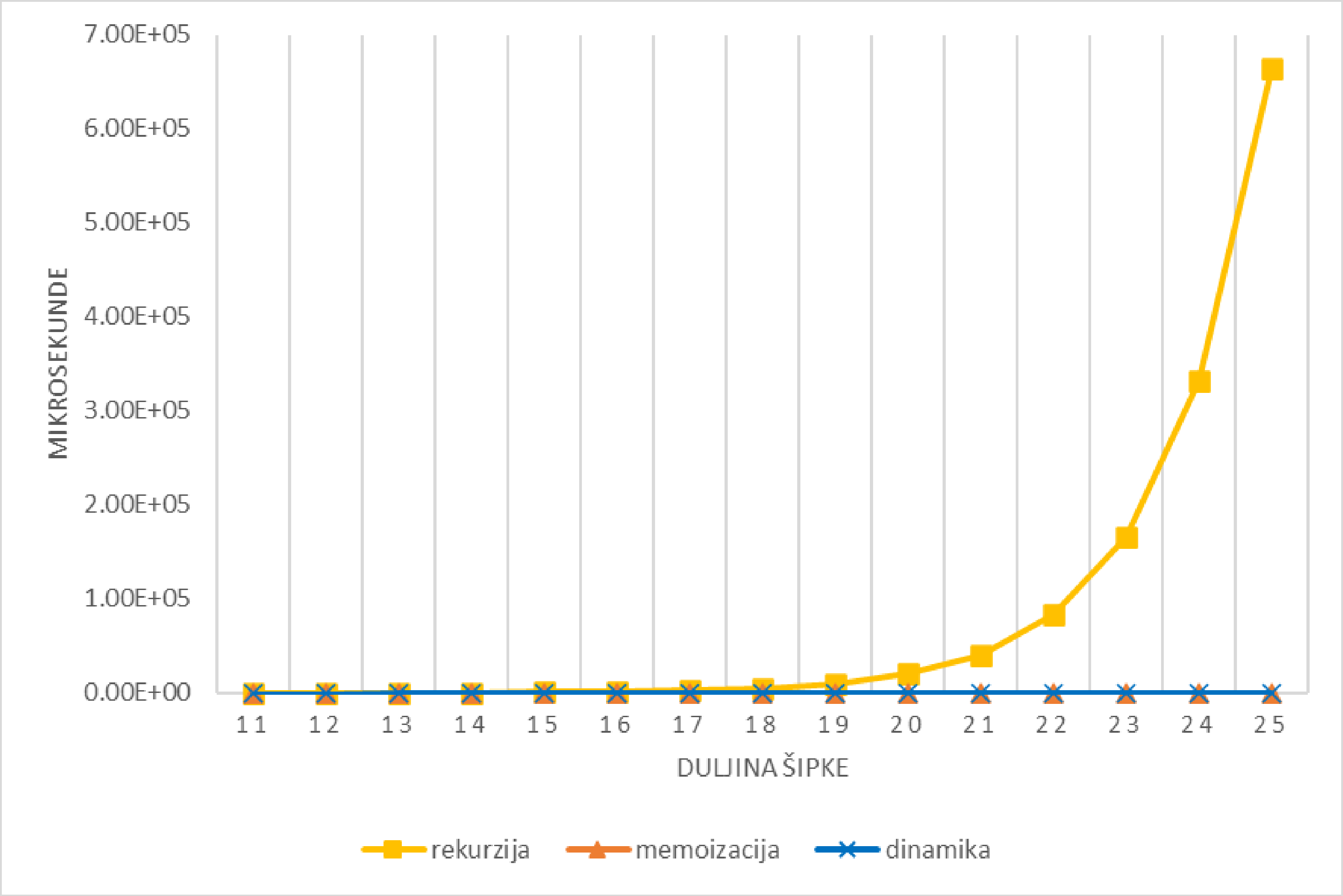
BOTTOM-UP RJEŠENJE I TEORIJSKA ANALIZA SLOŽENOSTI

- i7-1165 G7 2.80GHz
- 64GB RAM
- C++
- Stack size 4MB
- 100 mjerenja za pojedinu duljinu

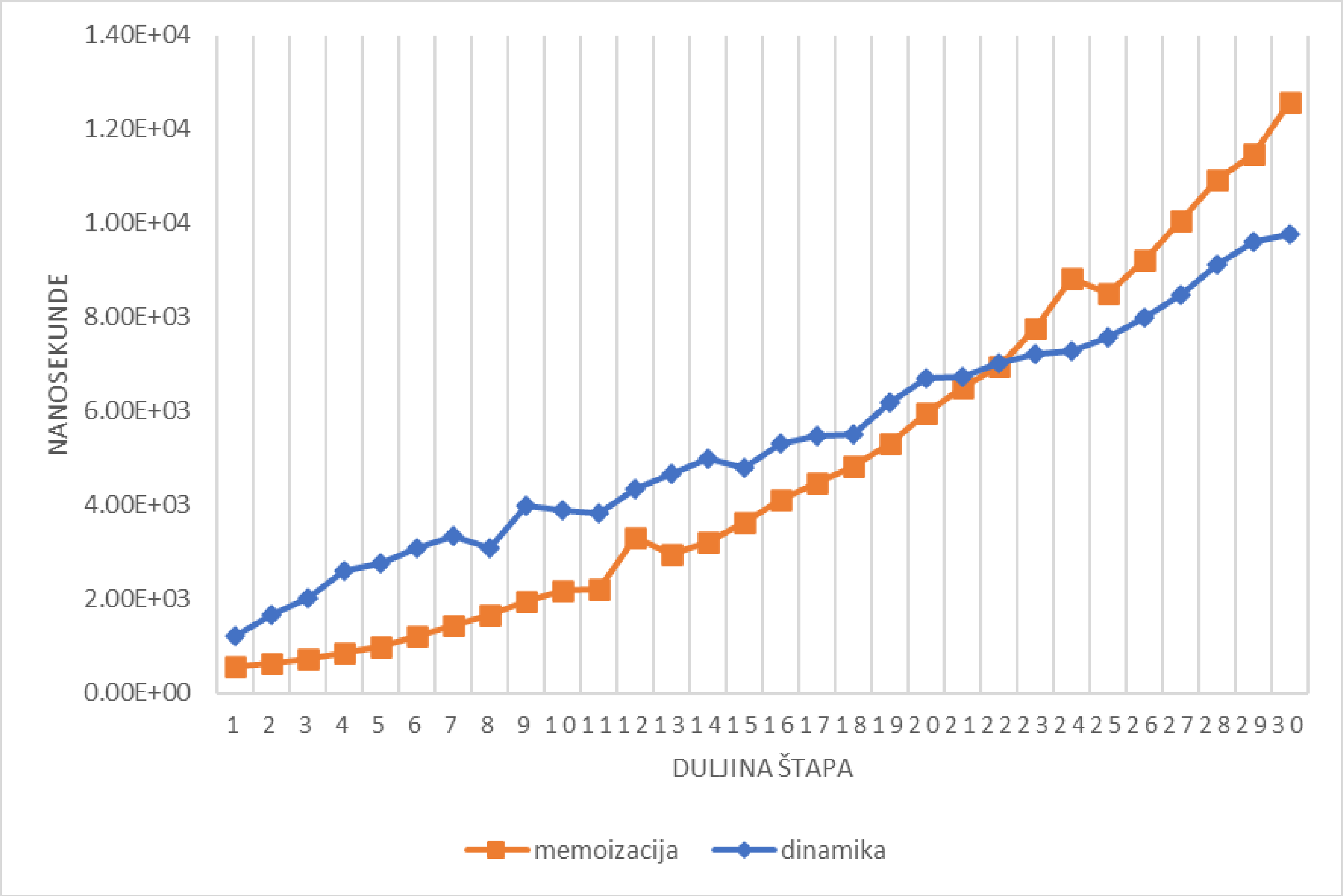
EMPIRIJSKA ANALIZA



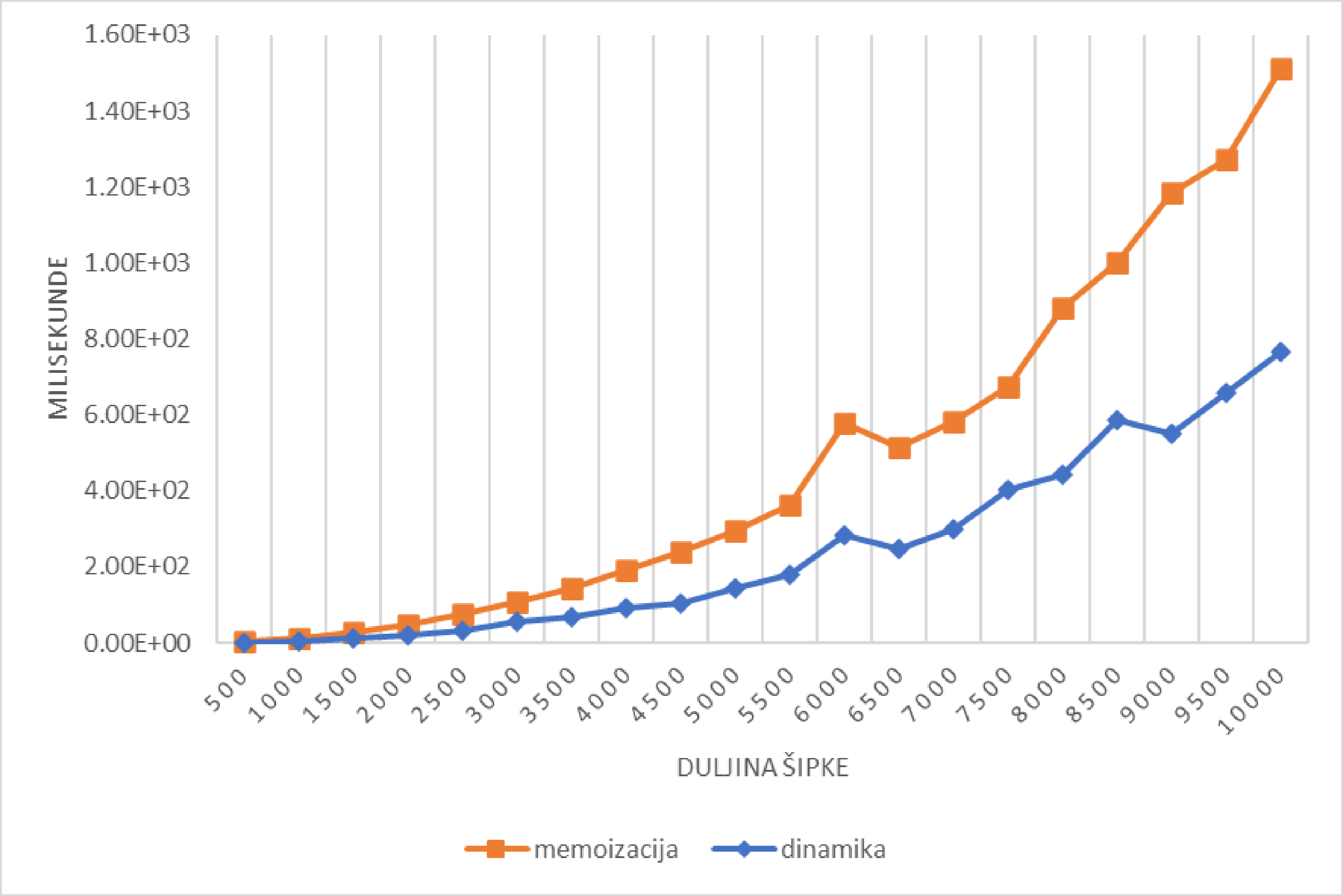
duljine od 1 do 10, povećanje duljine za 1, 100 ponavljanja, mjereno u nanosekundama



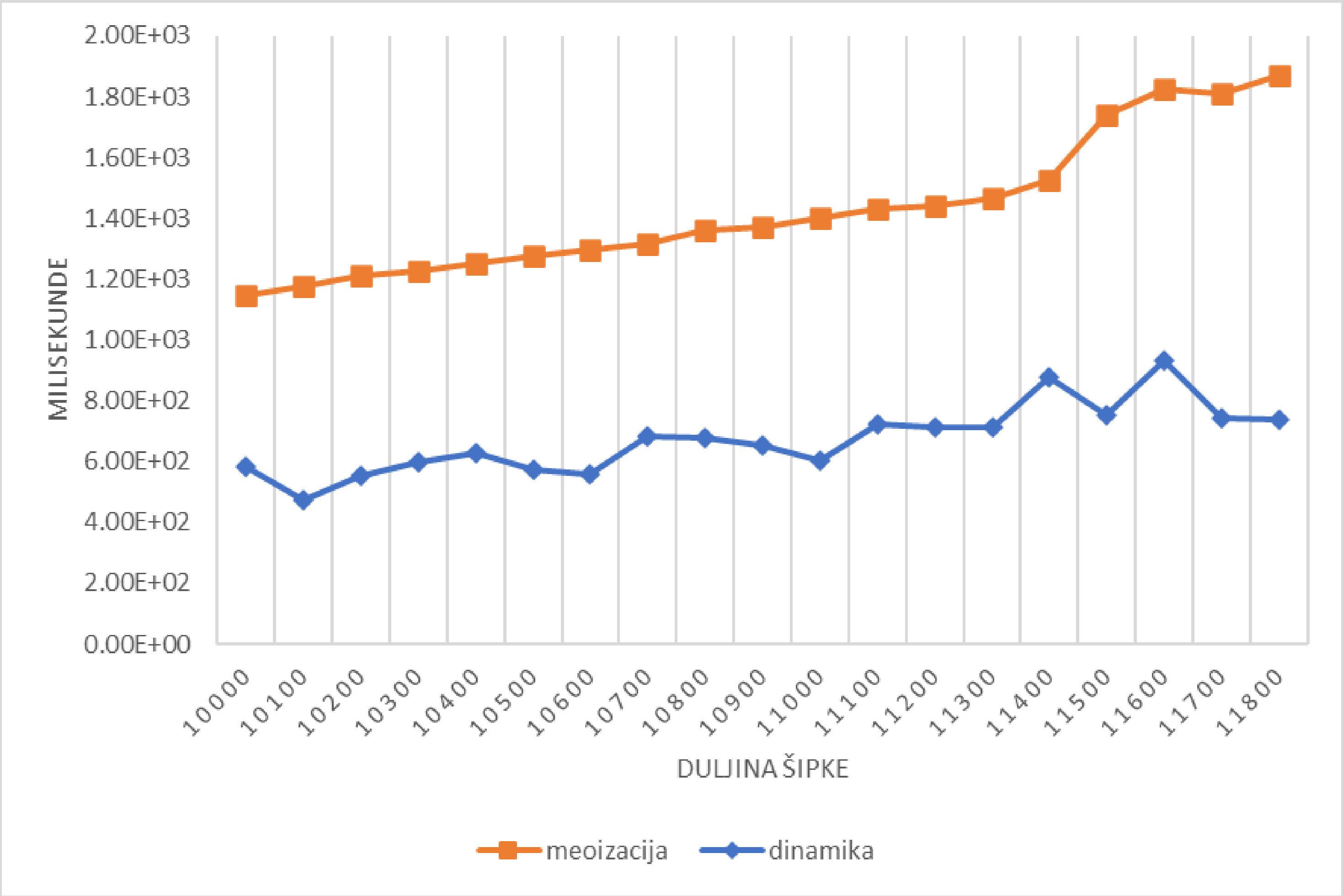
duljine od 11 do 25, povećanje duljine za 1, 100 ponavljanja, mjereno u mikrosekundama



duljine od 1 do 30, povećanje duljine za 1, 100 ponavljanja, mjereno u nanosekundama



duljine od 500 do 10 000, povećanje duljine za 500, 100 ponavljanja, mjereno u milisekundama



duljine od 10 000 do 11 800, povećanje duljine za 100, 3 ponavljanja, mjereno u milisekundama

IZVORI

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (Fourth Edition). MIT Press. https://mitpress.mit.edu/algorithms/
2. Mihelčić, M. 30.10.2023. "Dinamičko programiranje - nastavak, pohlepni algoritmi - uvod". kolegij Oblikovanje i analiza algoritama. Prirodoslovno - matematički fakultet, Sveučilište u Zagrebu https://web.math.pmf.unizg.hr/nastava/oaa/materijali/predavanjaMM/Predavanje6.pdf

HVALA NA POZORNOSTI