

Proiectarea Algoritmilor

Tema de casă

ANDjElOVIC ANDREJ

Calcuratoare română, anul I, 1.3 A

May 18, 2024

1 Enunțul problemei

1.1 Obiectiv

Asigurarea unei distribuții echitabile și eficiente a muncii între trei angajați care trebuie să găsească informații pe un raft de cărți, în special atunci când cărțile au dimensiuni diferite.

1.2 Metodă Simplă de Alocare

Dacă toate cărțile ar avea același număr de pagini, am putea împărți raftul în trei secțiuni egale. De exemplu:

- Fiecare secțiune cu 300 de pagini.

Această metodă funcționează bine cu cărți de dimensiuni egale.

1.3 Provocare cu Dimensiuni Variabile ale Cărților

Cu cărți de dimensiuni diferite, o împărțire simplă duce la volume de muncă inegale. De exemplu:

- Secțiunea 1: 600 de pagini
- Secțiunea 2: 1500 de pagini
- Secțiunea 3: 2400 de pagini

Acest lucru este clar inechitabil.

1.4 Alocare Strategică pentru Echitate

Pentru a echilibra volumul de muncă, putem realoca după cum urmează:

- Secțiunea 1: 1500 de pagini
- Secțiunea 2: 1300 de pagini
- Secțiunea 3: 1700 de pagini

Această aranjare asigură o distribuție mai echilibrată. Alocarea strategică a sarcinilor asigură echitate și eficiență, prevenind suprasolicitarea oricărui angajat.

2 Algoritmi

2.1 Adăugarea unei Noi Cărți într-o Listă Lantă

Explicație: Această funcție adaugă o nouă carte la o listă lantă de cărți.

Pseudo Cod:

```
adăugare_carte(carte_listă, nume, autor):  
    noua_carte = crează un nou nod de carte  
    setează numele noi_cărți la nume  
    setează autorul noi_cărți la autor  
    setează ID-ul bibliotecii noi_cărți la 0  
    setează pointerul următor al noi_cărți arate curent al listei de cărți  
    setează capul listei de cărți să arate către noua_carte
```

2.2 Afișarea Listei de Cărți

Explicație: Această funcție parcurge lista lantă de cărți și afișează detalii despre fiecare carte.

Pseudo Cod:

```
afișare_listă_cărți(carte_listă):  
    iterator = următorul nod al listei de cărți  
    cât timp iterator nu este NULL:  
        afișează numele iteratorului, autorul și ID-ul bibliotecii  
        mută iteratorul la următorul nod
```

2.3 Adăugarea unei Noi Biblioteci într-o Listă Lantă

Explicație: Această funcție adaugă o nouă bibliotecă la o listă lantă de biblioteci.

Pseudo Cod:

```
adăugare_bibliotecă(bibliotecă_colecție, ID):  
    noua_bibliotecă = crează un nou nod de bibliotecă  
    setează ID-ul noii_biblioteci la ID  
    setează pointerul noii_biblioteci arate către curentul listei de biblioteci  
    setează capul listei de biblioteci să arate către noua_bibliotecă
```

2.4 Afişarea Detaliilor Bibliotecii

Explicație: Această funcție parcurge lista lantă de biblioteci și afișează detalii despre fiecare bibliotecă.

Pseudo Cod:

```
afişare_detalii_bibliotecă(bibliotecă_colecție):  
    iterator = următorul nod al listei de biblioteci  
    cât timp iterator nu este NULL:  
        afișează ID-ul iteratorului  
        mută iteratorul la următorul nod
```

3 Rezultate Concluzii

În concluzie, atunci când ne confruntăm cu sarcina de a împărți încărcăturile de lucru între mai multe persoane, mai ales când avem de-a face cu dimensiuni sau complexități variate ale sarcinilor, alocarea strategică este crucială. După cum s-a demonstrat în scenariul sortării unei rafturi de cărți, o divizare simplă bazată pe secțiuni egale s-ar putea să nu fie suficientă atunci când dimensiunile cărților sunt diferite. În schimb, o distribuție echitabilă necesită o considerație atentă a factorilor variabili implicați. Prin utilizarea metodelor care vizează echilibrarea încărcăturii de muncă, cum ar fi gruparea elementelor de dimensiuni similare împreună, putem asigura echitatea și eficiența în alocarea sarcinilor. Acest lucru subliniază importanța adaptabilității și a gândirii strategice în gestionarea eforturilor colaborative, conducând în cele din urmă la rezultate optime pentru toate părțile implicate.