

## Tema 3 - Rope data structure

---

### Responsabili:

- Luca Istrate [mailto:lucaistrate@gmail.com]
- Dănuț Matei [mailto:matei.danut.dm@gmail.com]

**Data publicării : 13 mai, ora: 23:55**

**Deadline: 3 iunie, ora 23:55**

Modificări și actualizări

- **21 mai, ora 23:20** - actualizat secțiunea **Q&A** ca urmare a întrebărilor de pe **Forum**
- **30 mai, ora 16:30** - configurat tema pe **vmchecker**
- **30 mai, ora 19:42** - actualizat secțiunea **Q&A** ca urmare a întrebărilor de pe **Forum**

### Obiective

---

În urma realizării acestei teme:

- veți învăța să lucrați cu Arbori
- vă veți familiariza cu o structură de date folosită în editoare de text reale

### Introducere

---

Scopul acestei teme este implementarea unei structuri de date arborescente numită **Rope**. Aceasta permite efectuarea eficientă de operații (inserare, concatenare, ștergere etc.) pe un șir de caractere foarte lung (problemă caracteristică editoarelor de text de orice tip).

Un Rope este un arbore binar cu 2 *proprietăți*:

1. **Frunzele** conțin *stringuri* de lungime  $\geq 1$

2. Fiecare nod are o greutate asociată:

- pentru **frunze**, acest număr este *lungimea stringului* pe care îl conține fiecare
- pentru **restul nodurilor**, acest număr este *suma greutăților frunzelor din subarborele stâng*

Prin punerea cap la cap a stringurilor din frunze, de la stânga la dreapta, obținem stringul pe care îl reprezintă Rope-ul.

Înainte să vă apucați de implementare, citiți și secțiunile de **Precizări** și **FAQ**.

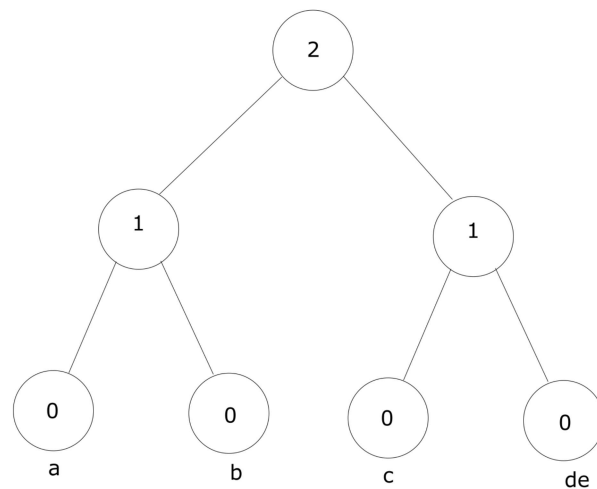
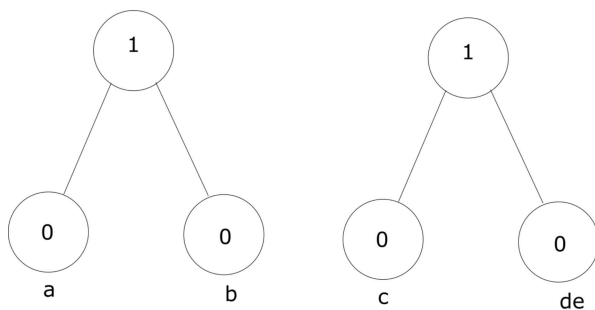
Va trebui să implementați următoarele operații:

#### 1. Concat - 10p

```
RopeTree* concat(RopeTree* rt1, RopeTree* rt2);
```

Concat(R1, R2) - concatenarea a 2 rope-uri

ex: Concat(Rope("ab"), Rope("cde")) → Rope("abcde")

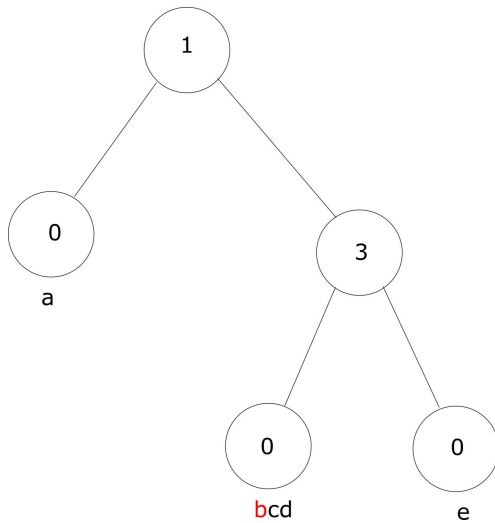


## 2. Index - 10p

```
char indexRope(RopeTree* rt, int idx);
```

Index(R1, i) - obținerea caracterului de pe poziția i dintr-un rope

ex: Index(Rope("abcde"), 1) → 'b'

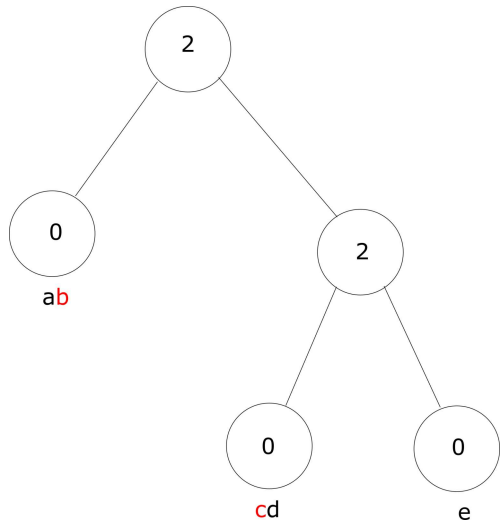


## 3. Search - 20p

```
char* search(RopeTree* rt, int start, int end);
```

Search(R1, start, end) - obținerea stringului din intervalul [start,end)

ex: Search(Rope("abcde"), 1, 3) → "bc"



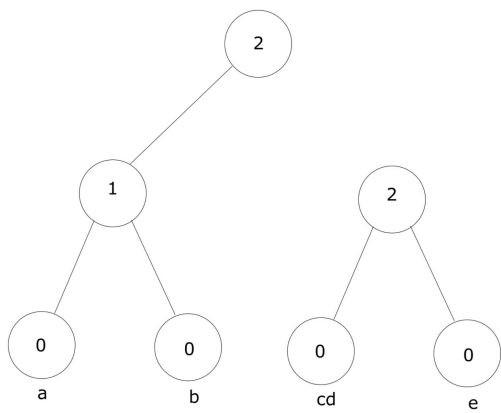
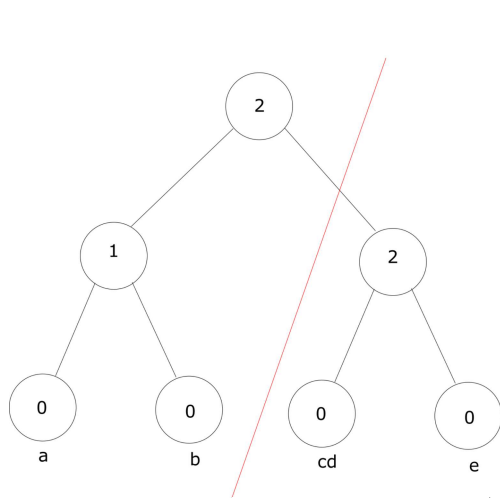
4. Split - 20p

SplitPair split(RopeTree\* rt, int idx);

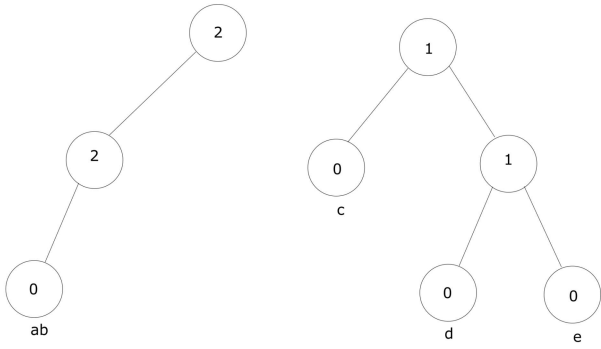
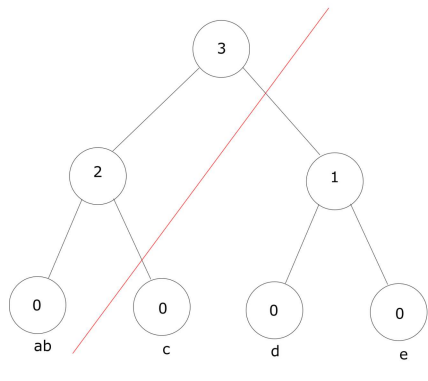
Split(R1, i) - împărțirea unui rope în 2 rope-uri separate

ex: Split(Rope("abcde"), 2) → (Rope("ab"), Rope("cde"))

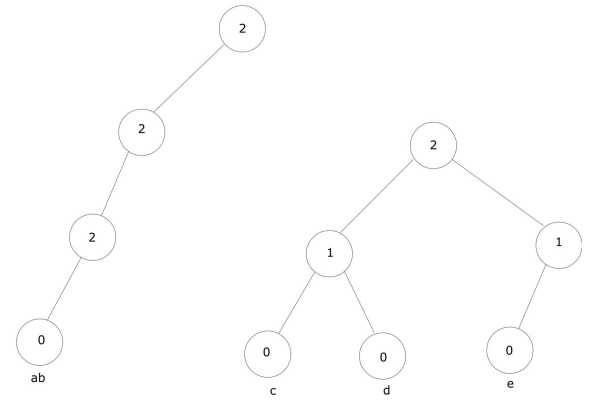
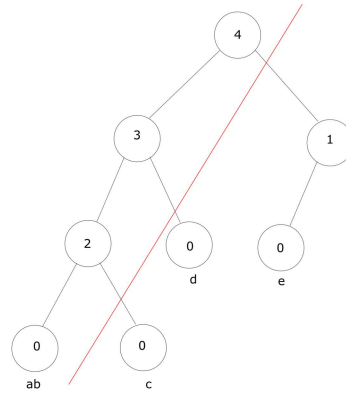
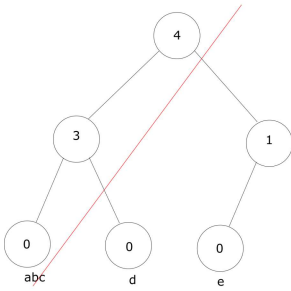
Exemplu 1:



Exemplu 2:



Exemplu 3:



## 5. Insert - 5p

```
RopeTree* insert(RopeTree* rt, int idx, const char* string);
```

Insert(R1, i, str) - inserarea unui string pe o poziție a unui Rope

ex: Insert(Rope("abcde"), 1, "123") → Rope("a123bcde")

Dacă ați implementat corect operațiile Split și Concat, Insert-ul va fi practic un Split, urmat de două Concat-uri.

## 6. Delete - 5p

```
RopeTree* delete(RopeTree* rt, int start, int len);
```

Delete(R1, start, len) - ștergerea substringului din intervalul [start, start+len)

ex: Delete(Rope("abcde"), 1, 2) → Rope("ade")

Dacă ați implementat corect operațiile Split și Concat, Delete-ul va fi practic două Split-uri, urmate de un Concat.

### Precizări

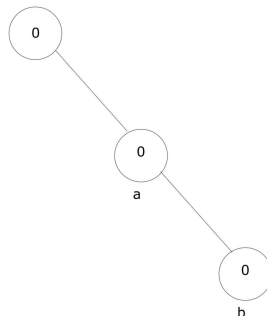
Aveți grijă ca Rope-urile create de funcțiile voastre să **respecte cele două proprietăți din definiția unui Rope!**

Nu încercați să vă folosiți de "trucuri" pentru a ocoli dificultatea cerințelor.

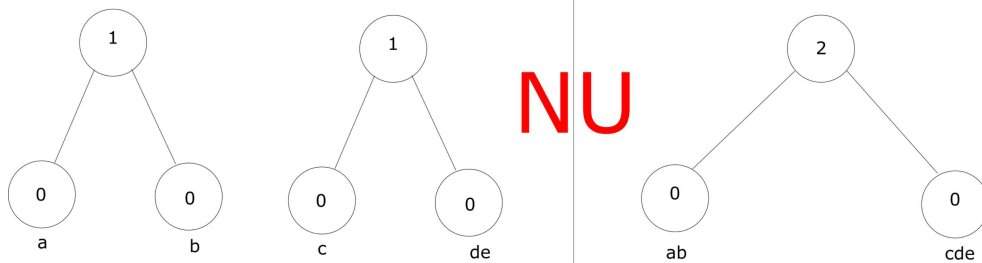
Următoarele sunt exemple de operații care, deși returnează Rope-uri care respectă definiția, **nu vor fi punctate**:

### Concat

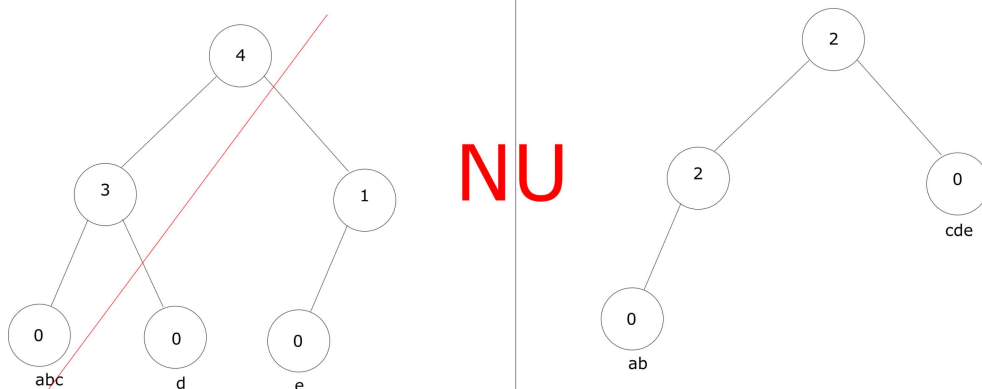
Dacă nu sunteți siguri, puteți întreba pe forum.



## Concat



## Split



## Schelet + Checker

## DOWNLOAD

Temele vor fi trimise pe vmchecker [<https://elf.cs.pub.ro/vmchecker/ui/#SD>]. **Atenție!** Temele trebuie trimise în secțiunea **Structuri de Date (CA)**.

Arhiva trebuie să conțină:

- surse
- fișier **README** care să conțină detalii despre implementarea temei

## Punctaj

**Atentie!** O temă care nu compilează va primi 0 puncte.

1. 80p teste
2. Testele pentru operațiile elementare sunt verificate cu valgrind. Dacă un test are memory leaks, nu va fi punctat.
3. 20p README + comentarii/claritate cod (ATENȚIE! Fișierul README trebuie făcut explicit, cât să se înțeleagă ce ați făcut în sursă, dar fără comentarii inutile și detalii inutile).
4. Se acordă 20% din punctajul obținut pe teste, ca bonus pentru coding style. De exemplu, pentru o temă care obține maxim pe teste, se pot obține 20p bonus dacă nu aveți erori de coding style. Pentru o temă ce trece 18 teste din 20, se pot obține 18p dacă nu aveți erori de coding style.
5. O temă care obține 0p pe vmchecker este punctată cu 0.
6. Temele au deadline hard. Prin urmare, o temă trimisă după deadline este punctată cu 0.

**Nu copiați!** Toate soluțiile vor fi verificate folosind o unealtă de detectare a plagiatului. În cazul detectării unui astfel de caz, atât plagiatorul cât și autorul original (nu contează cine e) vor primi punctaj 0 pe **toate temele!**

De aceea, vă sfătuim să nu vă lăsați rezolvări ale temelor pe calculatoare partajate (la laborator etc), pe mail/liste de discuții/grupuri etc.

## FAQ

**Q:** Pot modifica structurile din schelet?

**A:** Nu.

**Q:** Îmi pot implementa alte funcții helper, pe lângă cele cerute?

**A:** Desigur, chiar vă încurajăm să o faceți.

**Q:** Deci pentru Insert si Delete pot să dau copy-paste la codul din funcțiile Split si Concat?

**A:** Nu dați copy-paste, vă vom scădea din punctaj pentru cod duplicat. Pur și simplu chemați funcțiile deja implementate cu parametri corespunzători

**Q:** Rope-urile trebuie echilibrate de către noi?

**A:** Nu.

**Q:** Nu mă prind cum aș putea folosi funcția `getTotalWeight` din schelet. De ce ați inclus-o?

**A:** Sincer, am folosit-o în implementarea soluției și am uitat să o ștergem. Dacă nu vi se pare evident cum v-ați putea folosi de ea și mai mult vă încurcă, puteți să o ștergeți :)

**Q:** Am căutat informații despre Rope pe internet și am citit că greutatea unei frunze ar trebui să fie egală cu lungimea stringului pe care îl conține. Respectăm enunțul și o lăsăm 0? **A:** Vom considera ambele variante drept corecte.

**Q:** Aș vrea să folosesc un string auxiliar în care să îmi extrag tot șirul din Rope și să mă folosesc de el mai departe. E în regulă? **A:** Nu este permisă extragerea stringului din rope urmată de prelucrarea acestui string.

sd-ca/teme/tema3-2021.txt · Last modified: 2021/05/30 19:42 by gabriel\_danut.matei