1 - Random shuffling operacija nekog algoritma

Dat je niz operacija nekog algoritma. Želimo da random izmiješamo operacije tako da algoritam i dalje radi isto. Ovo se koristi kao tehnika obfuskacije kada želimo da osobe koje imaju pristup kodu ne mogu lagano shvatiti šta kod radi i mijenjati ga. Svaka operacija ima input varijable i output varijable.

Primjer:

```
a,b = Operacija(c, d, e)
d,f = Operacija(b, h)
y,g = Operacija(u, v)
b = Operacija(d, r)
```

Treća operacija se može izvršiti jednu ili dvije operacije ranije.

Postoji i petlja gdje se neki niz operacija može ponoviti određeni broj puta. U tom slučaju ne smijemo dopustiti da neka operacija van petlje uđe unutra ili neka unutra izađe van.

Primjer:

```
a,b = Operacija(c, d, e)
d,f = Operacija(b, h)
FOR_BEGIN(5)
        s = Operacija(d, k)
        d = Operacija(s, t)
FOR_END
y,g = Operacija(u, v)
b = Operacija(d, g)
```

2 - Raspored taskova na procesorima

Implementirati rješenje za https://codeforces.com/contest/1751/problem/A. Napraviti generisanje testnih podataka (na osnovu granica datih u problemu) i testirati više iteracija svog rješenja na njima.

3 - Generalizovana verzija 4. zadatka sa testa

Rječnik engleskih riječi skinuti sa interneta. Implementirati funkciju koja vraća random 100 parova riječi za koje treba k operacija (dovoljno je da radi za k<=6). Za full bodove implementirati optimizacije da radi brzo na velikom rječniku.

Zadatak sa testa:

> Dat je skup engleskih riječi. Amir želi da pravi zagonetke gdje su date dvije riječi i potrebno je postepeno promijeniti prvu u drugu nizom koraka tako da svaka riječ pripada skupu. U jednom koraku je moguće izbrisati jedno slovo trenutne riječi, dodati jedno slovo ili promijeniti jedno slovo. Svaka od operacija ima odgovarajuci trošak. Implementirati funkciju *vector<string> PromjenaSaNajmanjimTroskom(vector<string> rijeci, int indeks_pocetne, int indeks_krajnje, int trosak_dodavanja, int*

trosak_brisanja, int trosak_zamjene) koja vraća sve riječi koje ucestvuju u promjeni u pravom poretku. Ukoliko ima više mogućih promjena sa najmanjim troškom vratiti bilo koju.

PromjenaSaNajmanjimTroskom({“eat”, “treat”, “meat”, “bar”, “far”, “ear”, “beat”, “bear”}, 0, 3, 1, 1, 1) -> {“eat”, “ear”, “bar”}
PromjenaSaNajmanjimTroskom({“eat”, “treat”, “meat”, “bar”, “far”, “ear”, “beat”, “bear”}, 0, 3, 1, 1, 3) -> {“eat”, “ear”, “bear”, “bar”}

4 - Predikcija konačnog rezultata tenis meča

Teniski meč se igra i na osnovu trenutnog rezultata želimo odrediti vjerovatnoću da će određeni igrač pobijediti. Pretpostavimo da imamo brojeve p1 i p2 koji predstavljaju vjerovatnoću da će igrač 1 osvojiti poen na svom servisu i vjerovatnoću da će igrač 2 osvojiti poen na svom servisu u svakom poenu do kraja igre. Možete prvo pretpostaviti da imate ove vjerovatnoće kao input. Kasnije možete sami naći aproksimaciju te vjerovatnoće (npr. u odnosu na to kako je igrač igrao na svom servisu do sad u meču ili generalno koliko je taj igrač dobar) te testirati svoj algoritam na stvarnim mečevima.

5 - Zadaci sa codeforces.com

Riješiti 10 zadataka sa tagom dp i težinom barem 1800. Nakon što pokušate uraditi svaki zadatak (predlažem sat vremena da probate) možete vidjeti ideju sa tutorijala i onda implementirati sami. Na odbrani ću ja izabrati zadatak čije rješenje trebate objasniti. Trebate osigurati da nikoja dva studenta ne uzmu isti zadatak.

6 - Neka vaša ideja

Trebate napisati prijedlog projekta. Ne mora biti dug prijedlog, ali mora biti jasno šta se planira uraditi.