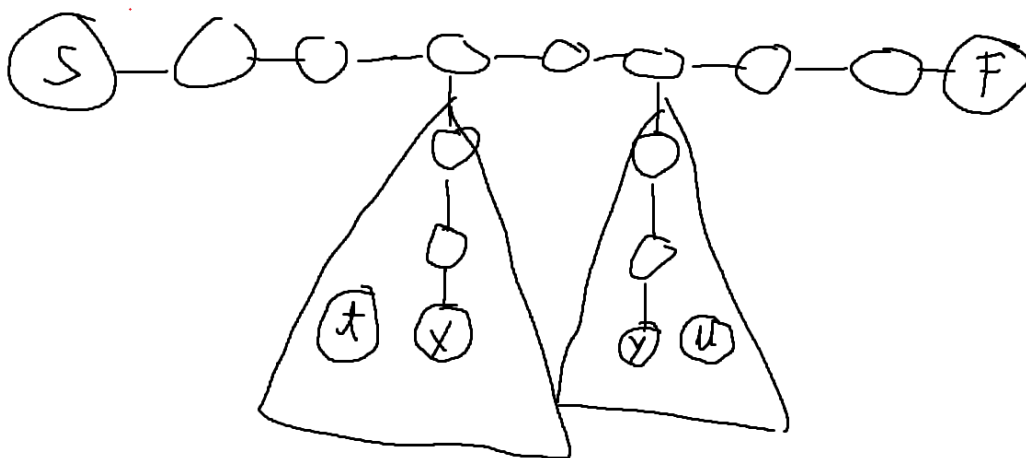


Vom incepe cu cazul cand drumul  $x$   $y$  se intersecteaza cu drumul  $s$   $f$ . Atunci, ne putem imagina ca arborele ar arata ca in desenul de mai jos, iar perechile  $(t,u)$  valide ar fi toate care respecta urmatoarele proprietati:

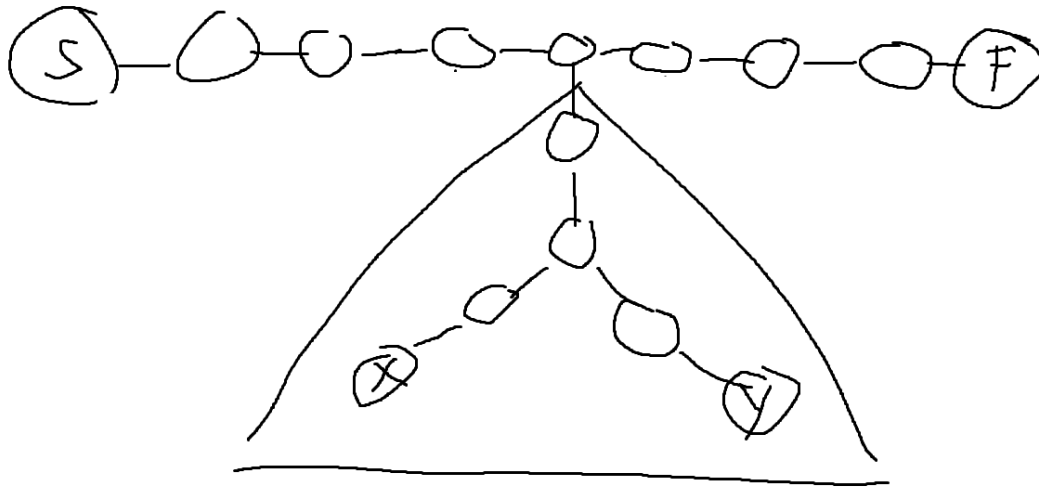
- $t$  se afla in acelasi subarboare ca  $x$  (sau mai formal spus, drumul de la  $t$  la  $x$  nu se intersecteaza cu drumul de la  $s$  la  $f$ )
- $u$  se afla in acelasi subarboare ca  $y$
- $\text{dist}(x,t) + \text{dist}(y,u) + 1 < \text{dist}(x,y)$



Dar stim ca suma din  $\text{dist}(x,y) \leq 1000000$  pentru toate query-urile asa ca putem sa fixam  $\text{dist}(x,t)$  si apoi am vrea sa numaram (cate noduri din acelasi subarboare cu  $x$  la distanta  $\text{dist}(x,t)$  de  $x$ ) \* (cate noduri din acelasi subarboare cu  $y$  la distanta  $< (\text{dist}(x,y) - \text{dist}(x,t) - 1)$  de  $y$ ).

Pentru a putea sa calculam rapid cate astfel de noduri exista, intai o sa vedem pentru toate query-urile, care sunt toate aceste lucruri care trebuie calculate iar apoi o sa le calculam pe toate de odata, folosindu-ne de faptul ca avem query-urile offline. Mai exact, o sa ne retinem perechi  $(x,d)$  cu semnificatia ca: "vreau sa calculez cate noduri din subarboarele lui  $x$  se afla la distanta  $\leq d$  de nodul  $x$ ". Datorita faptului ca suma din  $\text{dist}(x,y) \leq 1000000$ , vom avea cel mult 2000000 astfel de pair-uri care trebuie calculate. Le vom calcula prin centroid decomposition, intrucat toate requirement-urile astea intreaba cate noduri sunt in cate un anumit subarboare, putem rezolva separat pentru fiecare subarboare. Sa zicem ca suntem intr-un anumit subarboare si vrem sa rezolvam toate perechile  $(x,d)$  din acesta, vom folosi centroid decomposition. Fixam centroidul, si adaugam la fiecare pair contributia tuturor nodurilor pentru care drumul trece prin centroid, doar facem cate un dfs din centroid in vecinii lui, retinem cate noduri se afla la fiecare adancime fata de el, iar apoi ne folosim de acestea pentru a contribui la toate pair-urile cu valorile corespunzatoare. Astfel, cu centroid decomposition putem sa precalculez toate valorile de care vom avea nevoie atunci cand vrem sa calculam raspunsul pentru un anumit query.

In cazul cand drumul  $x, y$  nu se intersecteaza cu drumul  $s, f$ , arborele  $o$  sa arate similar cu ce e in desenul de mai jos



De data asta, o sa numaram cate perechi  $(t, u)$  nu sunt valide. Astfel, ele trebuie sa respecte urmatoarele proprietati:

- sau  $t$  sau  $u$  nu se afla in subarbore
- $\text{dist}(x, t) + \text{dist}(y, u) + 1 < \text{dist}(x, y)$

Din nou putem sa fixam  $\text{dist}(x, t)$  iar apoi sa numaram cate perechi sunt valide folosindu-ne de ce am precalculat pentru primul caz. Totusi, aici mai avem nevoie si sa raspundem la intrebari de forma: "cate noduri sunt la distanta  $\leq d$  de  $x$  (in tot arborele)". Dar astea se pot calcula intr-un mod similar cu ce am facut in primul caz, retinand perechile  $(x, d)$  care trebuie calculate si aflandu-le cu centroid decomposition.