

1. Реализация `man_slow.cpp`:

Поддържам масив `id[]`, притежаващ клетка за всеки компютър, със свойството, че елементите `id[p]` и `id[q]` са свързани точно тогава, когато компютри `p` и `q` са свързани в мрежа. В този случай, за да се свържат `N` компютри са необходими `M` обединения, следователно сложността на алгоритъма е: $O(MN)$.

2. Реализация `man.cpp`:

Използвам отново масив `id[]`, притежаващ клетка за всеки компютър. Като всеки компютър сочи към друг компютър от същото множество, в структура без цикли. За да определя дали два компютъра са в едно и също множество следвам указателите за всеки от тях, докато стигна до компютър, който сочи себе си. Компютрите са в едно и също множество тогава и само тогава, когато този процес ги отвежда към един и същи компютър. Ако не са в същото множество, стигам до различни обекти. За да извърша обединението, просто свързвам единия компютър към другия. Когато обработвам двойката `p` и `q`, следвам указателите от `p`, за да стигна до стойност `i` с `id[i]==i`, тогава следвам указатели от `q`, за да стигна до стойност `j` с `id[j]==j`. След това, ако `i` и `j` се различават едно от друго, задавам `id[i]=id[j]`. За да подобря ефективността на алгоритъма използвам още едним масив `v[]` за всеки елемент `id[i]==i`, който съдържа броя на върховете в съответното дърво (дървото представлява структура от свързани вече компютри), така че обединението може да свързва по-малкото от две указани дървета с по-голямото, като по този начин се избягва създаването на дълги пътища в дърветата. Сложността на този алгоритъм е: $O(M \cdot \lg N)$.