

IMPA Omgång 2 – 2010-09-15

10670 – Work Reduction

Girig.

```
För varje arbetare
  Repetera tills korrekt arbetsnivå
    m = Nuvarande arbetsbelastning
    c1 = Halveringskostnad
    c2 = ceil(n/2)*Reduceringskostnad
        (Alltså: Halvering genom många reduceringar)
    Om c1 < c2 och halvering är möjlig
      Halvera
    Annars
      Reducera
Sortera
```

910 – TV Game

Dynamisk programmering. Maximalt 30 steg ger maximalt 30^2 kombinationer av vägar vilket får plats i en 32-bitars integer.

Svaret blir helt enkelt $\text{WaysToNode}(\text{node}, \text{moves}) = \sum_{n=\text{input nodes}} \text{WaysToNode}(n, \text{moves} - 1)$.

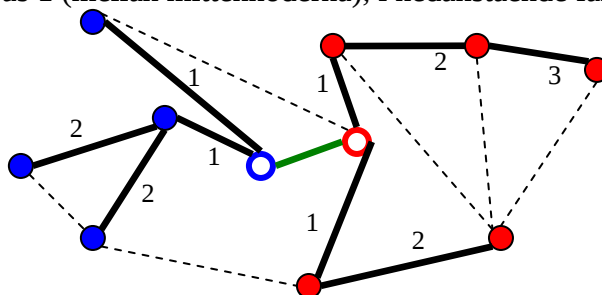
Använd memoization eller skriv n riktig DP:

```
for m = 1..moves
  for each node n
    waysToNode[m][destination0[n]] += waysToNode[m-1][n]
    waysToNode[m][destination1[n]] += waysToNode[m-1][n]
```

10805 – Cockroach Escape Networks

Minimum diameter spanning tree.

Välj två noder och ansätt som mitten. Hitta kortaste väg för alla noder till någon av mittennoderna. Diametern i trädet är summan av längsta vägarna d_i inom de två subträden till respektive mittpunkt plus 1 (mellan mittennoderna), i nedanstående fall $2+3+1$.



Hur kommer det sig att detta funkar? Jo, antag att man med algoritmen kommit fram till svaret x . Välj två noder. Om de finns i olika subträd så har algoritmen redan löst uppgiften. Om noderna finns i samma träd så kan vägen vara så lång som d_i till mittennoden dvs. total väg är $d_i + d_i \leq d_1 + d_2 + 1$ om de två noderna som man valde i början verkligen var i mitten, alltså $|d_1 - d_2| \leq 1$.

Vilka mittennoder ska man välja? Det finns en algoritm för absolute 1-center, men i detta fall går det att prova alla par av två noder intill varandra.