1)  
a)  
 for k ∈ [1 : m] do: // initializare

C[k] = ∅; W(K)= 0;

for i ∈ [1 : n] do: // pentru fiecare cadou ‘i’ ; O(n)

aleg q indicele copilului cu W(q) minim. // aceasta alegere este O(1) daca folosim un priority queue pentru valorile cadourilor primite de copii

C[q]=C[q]U(i)

W(q)+=val(i) //O(log m) - inserare un element in priority queue

b) Deoarece cadoul *i* este atribuit lui *q*, inseamna ca la acel moment *q* avea o valoare a cadourilor minima dintre toti copiii la acel moment, deci <= decat k la acel moment

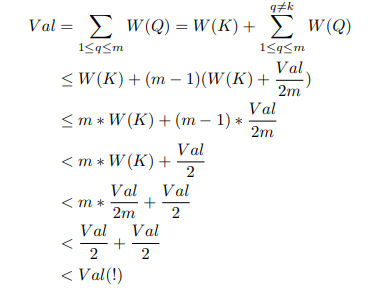
deci W(k)>=W(q)-val(i)

c) fie k acel copil pt care W(k)=ALG. presupunem ca W(k)<Val/(2m):

stim ca W(k)+val(i)>=W(q) pentru orice alt copil q si *i* ultimul cadou primit de q

si din ipoteza stim ca val(i)<Val/2m

avem:



d) trebuie sa aratam ca 1/2OPT<=ALG

Upper bound pentru OPT?

OPT<=Val/m

din punctul c) am aratat ca ALG>=Val/2m

1/2OPT<=val/(m\*2)<=ALG

2)

a)

tb sa minimizam ILP cu restrictiile  
 - suma task-urilor pe fiecare masina sa nu depaseasca ILP, matematic:  
 pentu fiecare masina q