

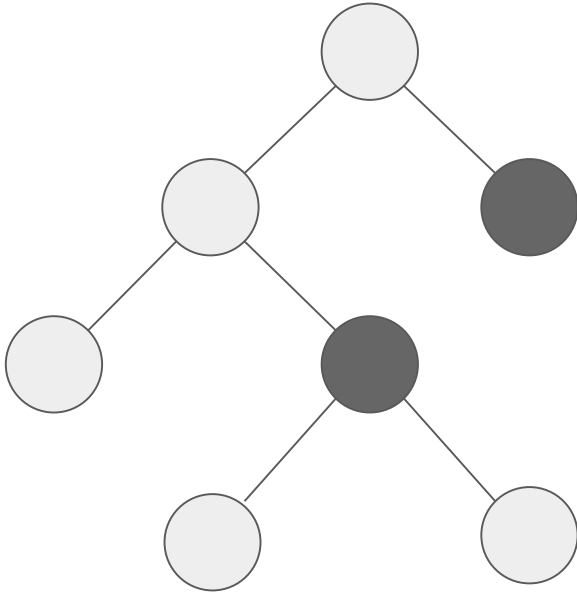
Example of Tree Tiling using Dynamic Programming

Compilers Course
MIEIC
2018

The Problem

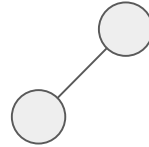
- Given an input tree and a set of subtrees (each one with an associated cost) determine the tree tiling with minimum cost
- Assumptions: the input tree can be fully tiled with the considered set of subtrees
- Connection with the Instruction selection problem:
 - Input tree is a tree-based low-level IR of a code statement (or region of code)
 - Set of subtrees represents the IR tree patterns
 - Cost of each subtree represent the cost of each instruction (e.g., execution clock cycles). If the goal is to find the tree tiling with the minimum number of subtrees, one case use a cost of 1 for each subtree

Input Tree

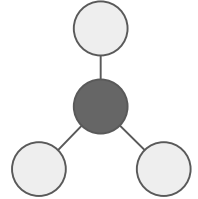


Sub Trees

(A) Cost: 2



(B) Cost: 5



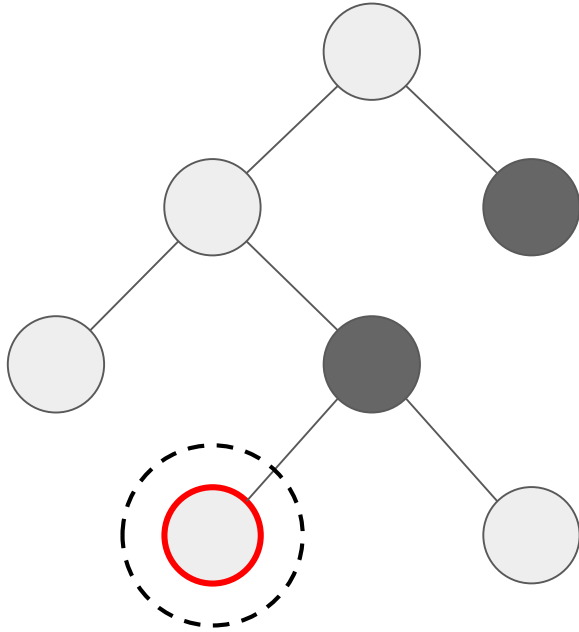
(C) Cost: 4



(D) Cost: 3



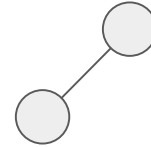
Input Tree



Sub Trees

NO MATCH

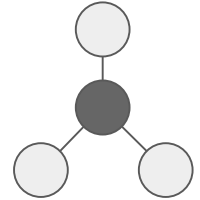
(A) Cost: 2



(C) Cost: 4



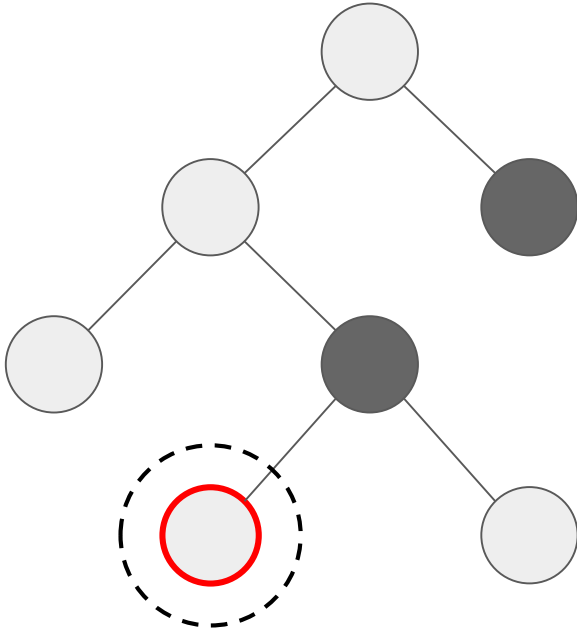
(B) Cost: 5



(D) Cost: 3

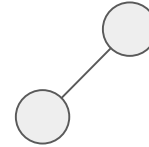


Input Tree



Sub Trees

(A) Cost: 2

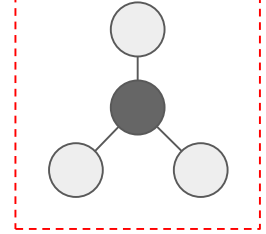


(C) Cost: 4



NO MATCH

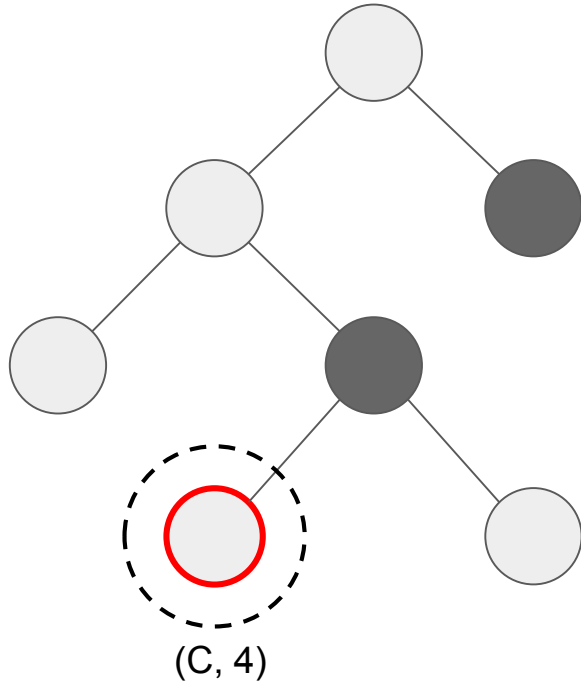
(B) Cost: 5



(D) Cost: 3

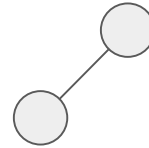


Input Tree

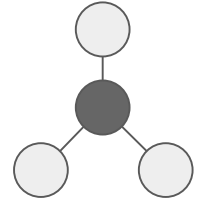


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

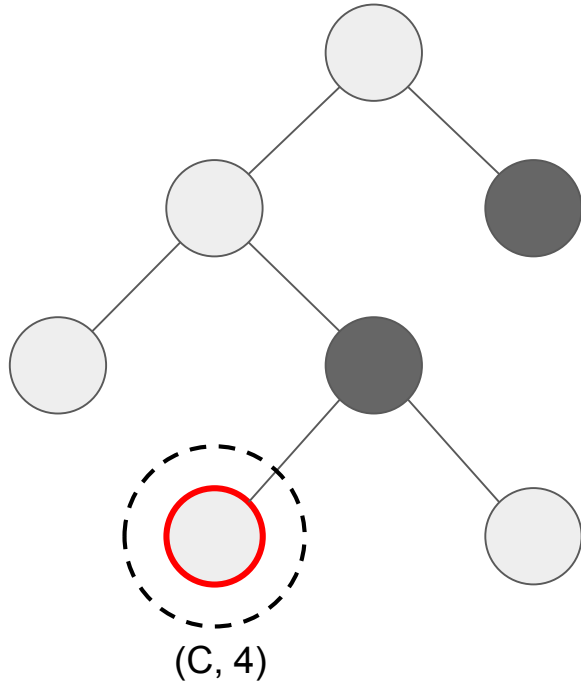
(C) Cost: 4



(D) Cost: 3

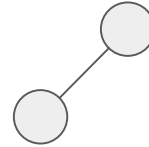


Input Tree

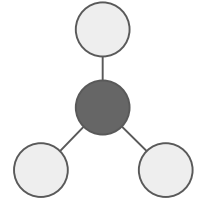


Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4

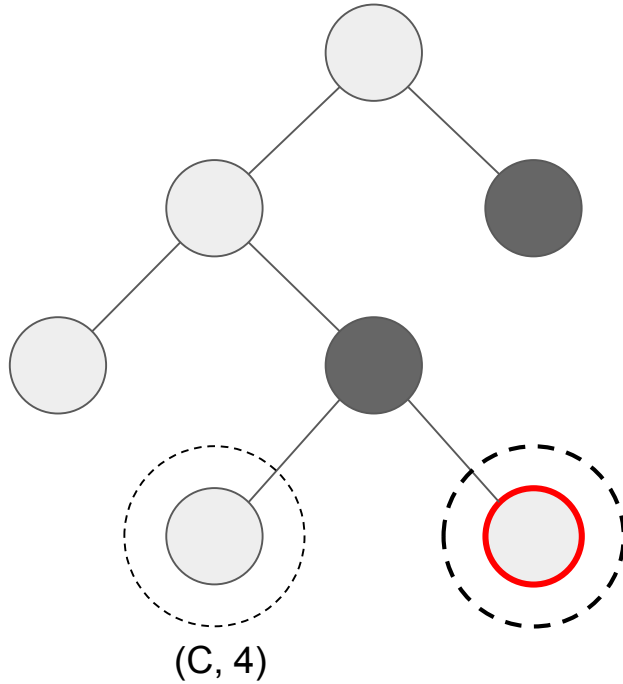


NO MATCH

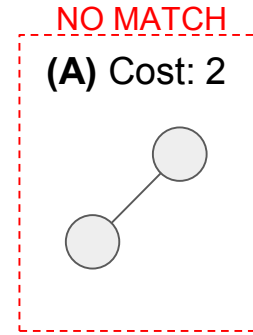
(D) Cost: 3



Input Tree



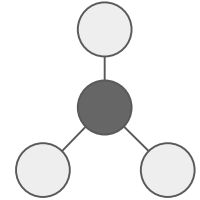
Sub Trees



(C) Cost: 4



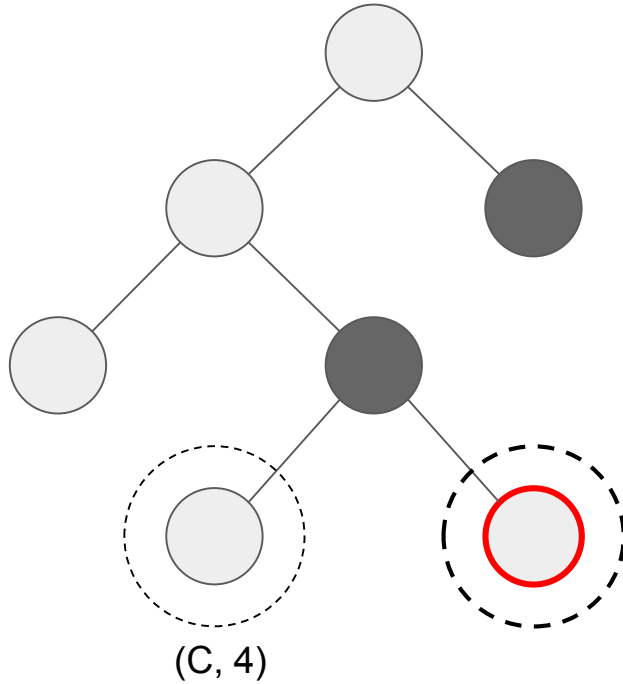
(B) Cost: 5



(D) Cost: 3

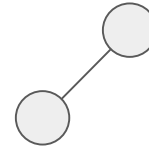


Input Tree



Sub Trees

(A) Cost: 2

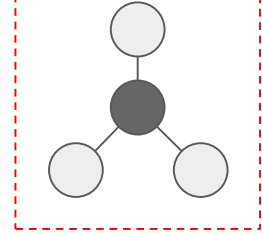


(C) Cost: 4



NO MATCH

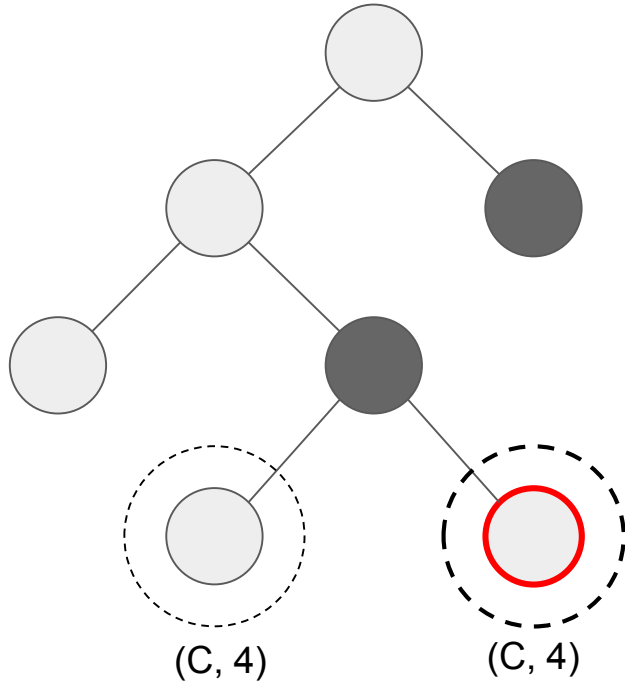
(B) Cost: 5



(D) Cost: 3

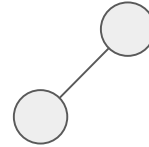


Input Tree

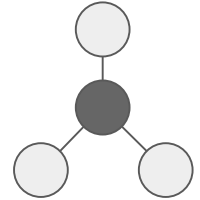


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

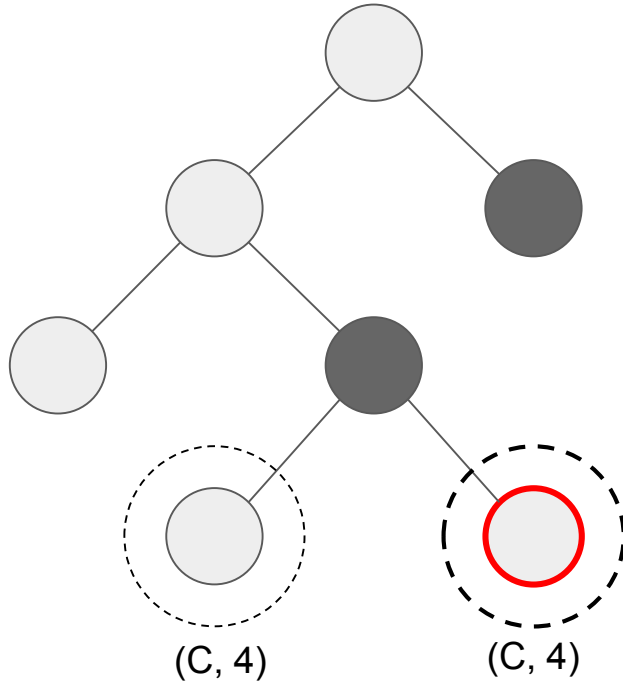
(C) Cost: 4



(D) Cost: 3

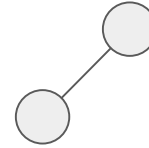


Input Tree

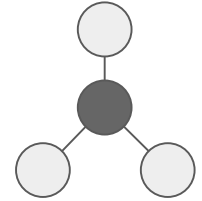


Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4

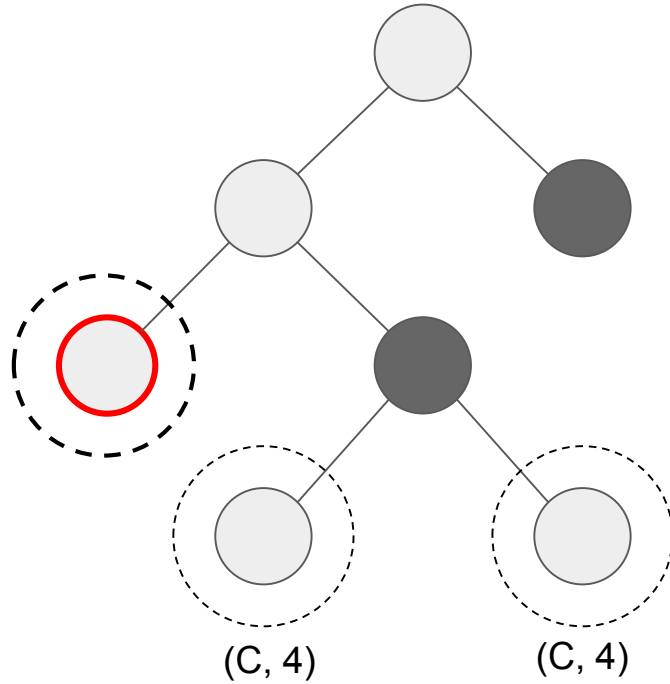


NO MATCH

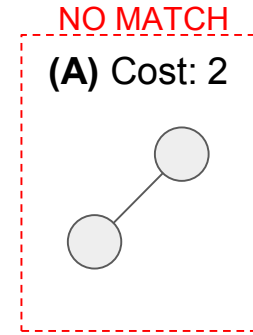
(D) Cost: 3



Input Tree



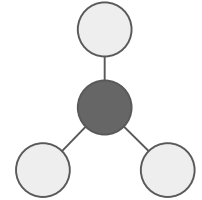
Sub Trees



(C) Cost: 4



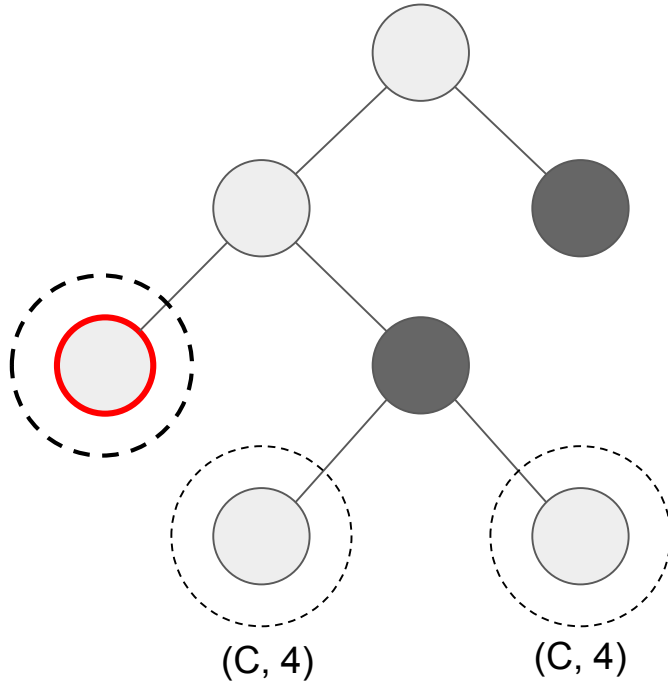
(B) Cost: 5



(D) Cost: 3

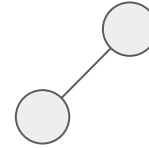


Input Tree



Sub Trees

(A) Cost: 2

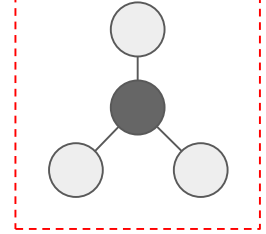


(C) Cost: 4



NO MATCH

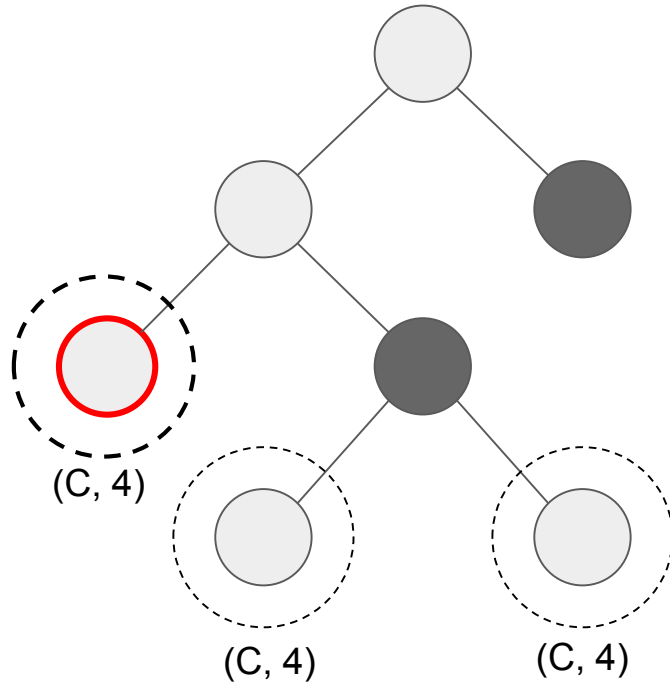
(B) Cost: 5



(D) Cost: 3

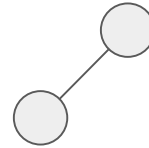


Input Tree

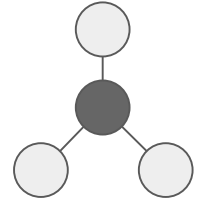


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

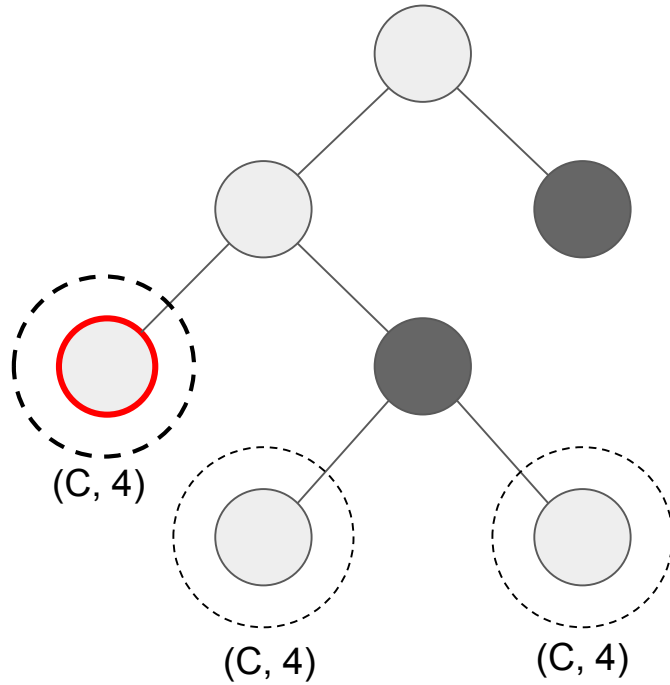
(C) Cost: 4



(D) Cost: 3

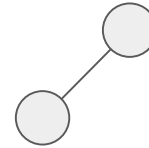


Input Tree

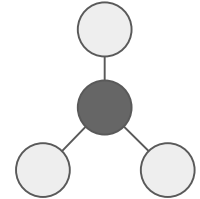


Sub Trees

(A) Cost: 2



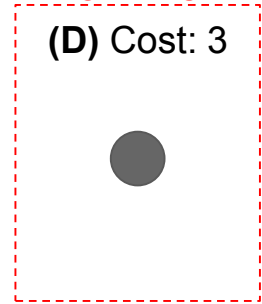
(B) Cost: 5



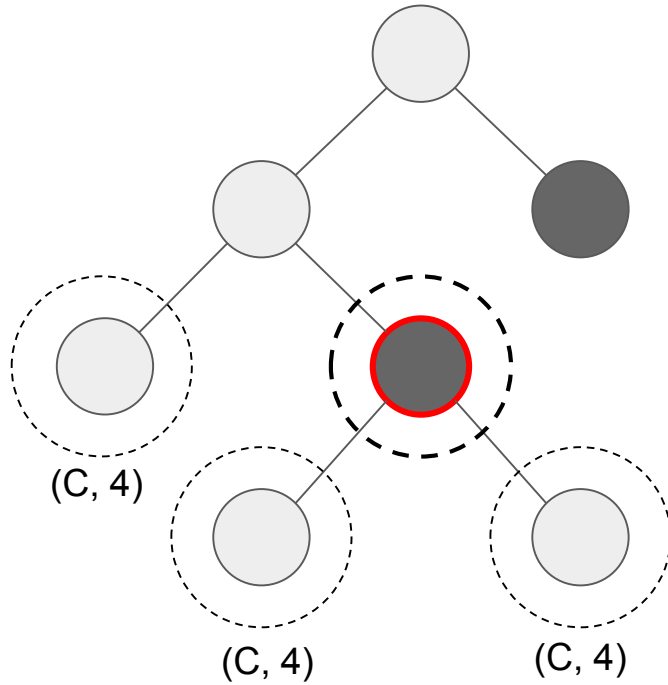
(C) Cost: 4



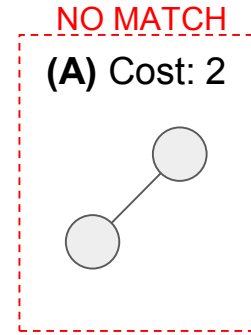
NO MATCH
(D) Cost: 3



Input Tree



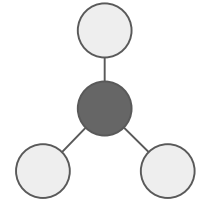
Sub Trees



(C) Cost: 4



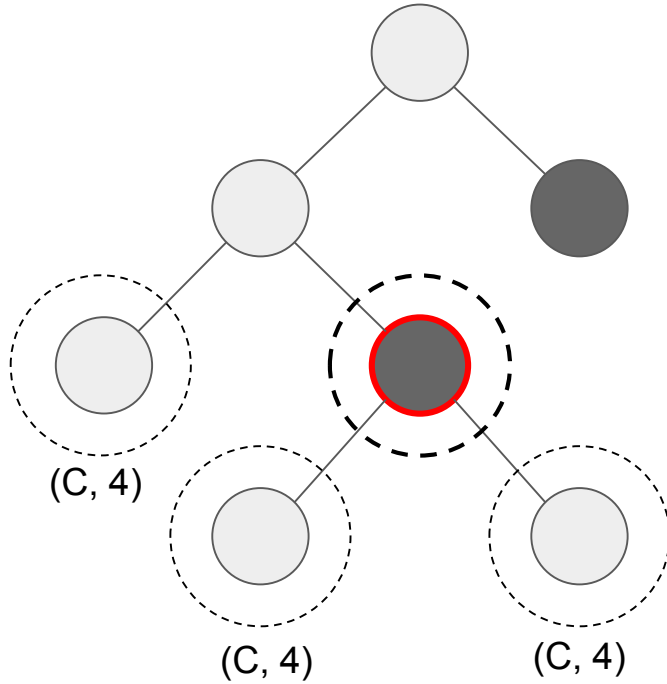
(B) Cost: 5



(D) Cost: 3

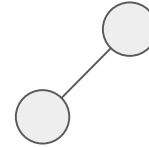


Input Tree



Sub Trees

(A) Cost: 2

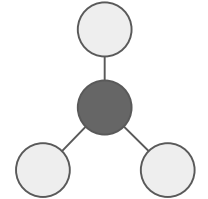


(C) Cost: 4



NO MATCH

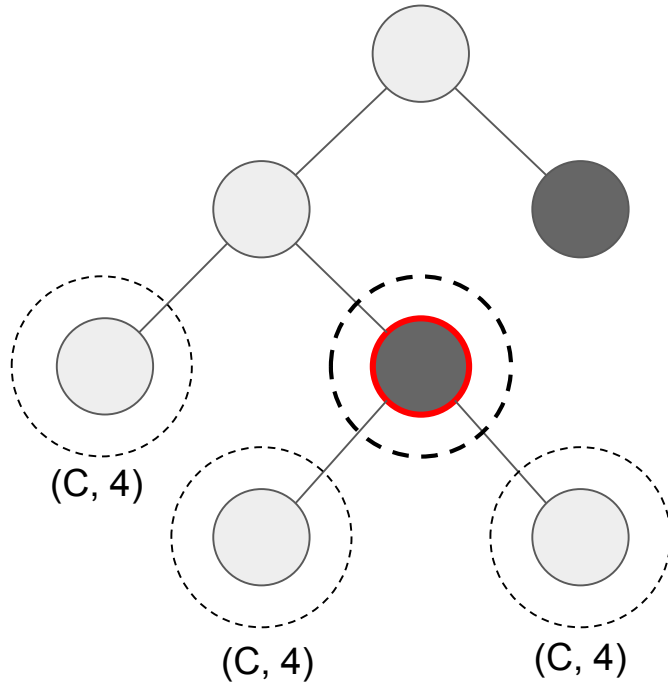
(B) Cost: 5



(D) Cost: 3

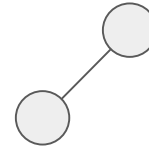


Input Tree

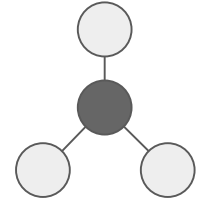


Sub Trees

(A) Cost: 2



(B) Cost: 5



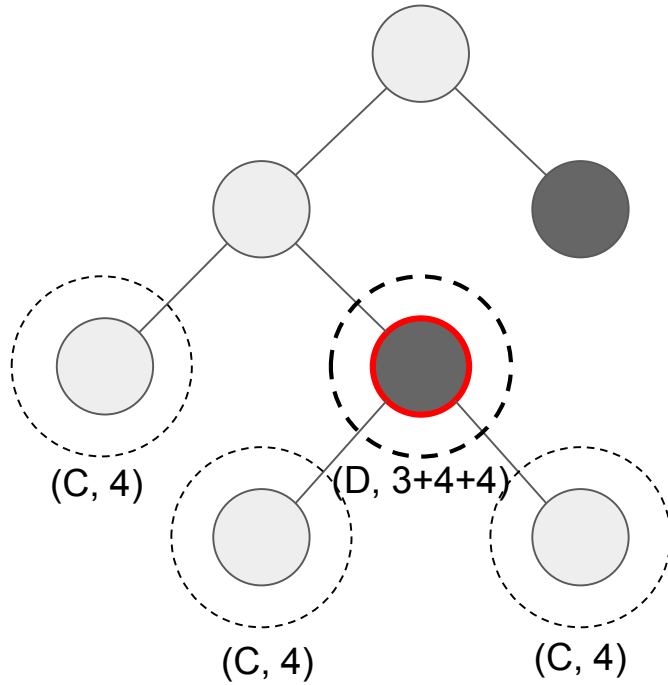
NO MATCH
(C) Cost: 4



(D) Cost: 3

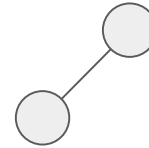


Input Tree

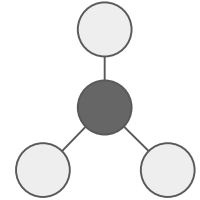


Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4

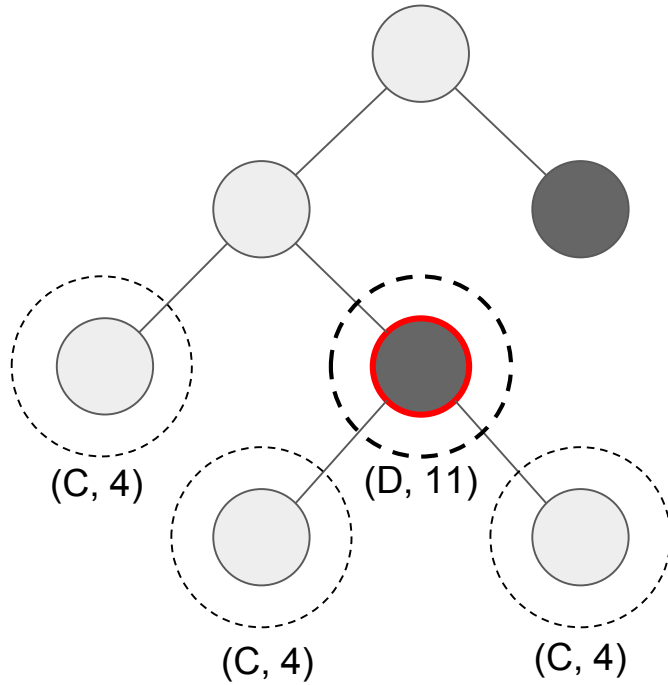


MATCH

(D) Cost: 3

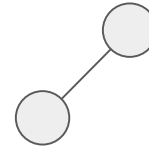


Input Tree

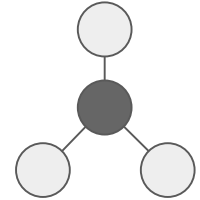


Sub Trees

(A) Cost: 2



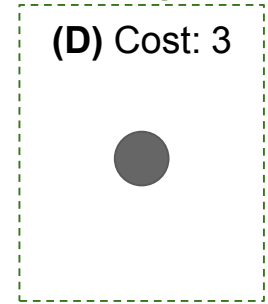
(B) Cost: 5



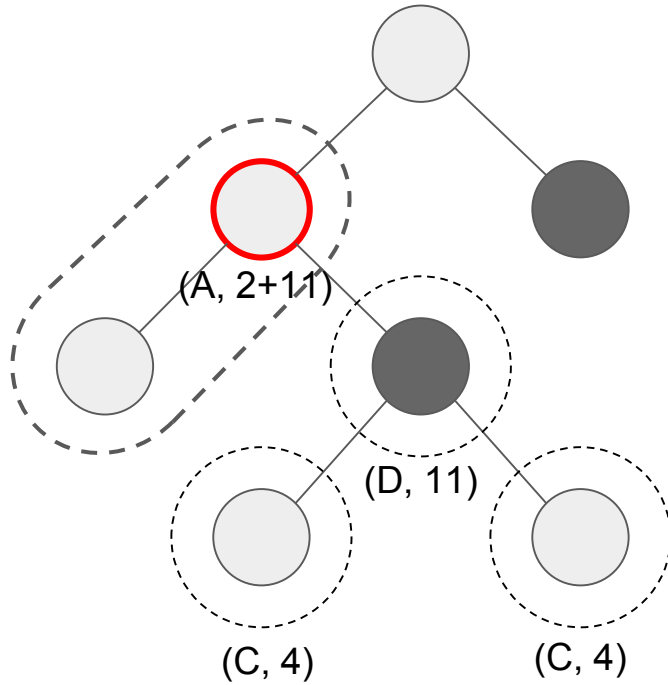
(C) Cost: 4



MATCH
(D) Cost: 3



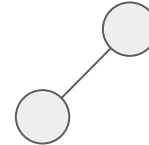
Input Tree



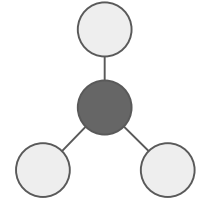
Sub Trees

MATCH

(A) Cost: 2



(B) Cost: 5



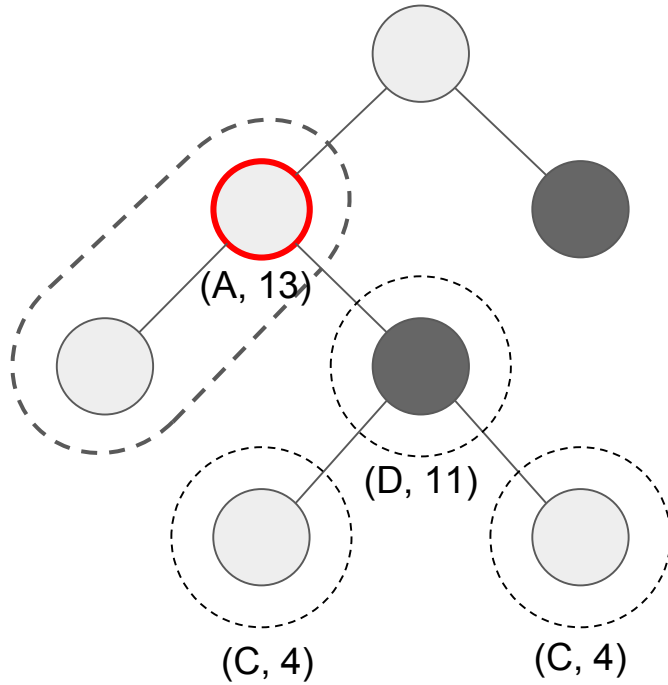
(C) Cost: 4



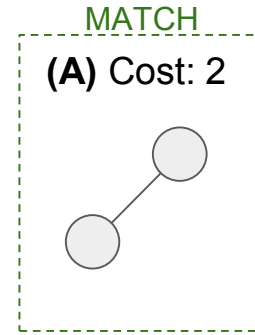
(D) Cost: 3



Input Tree



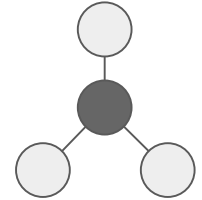
Sub Trees



(C) Cost: 4



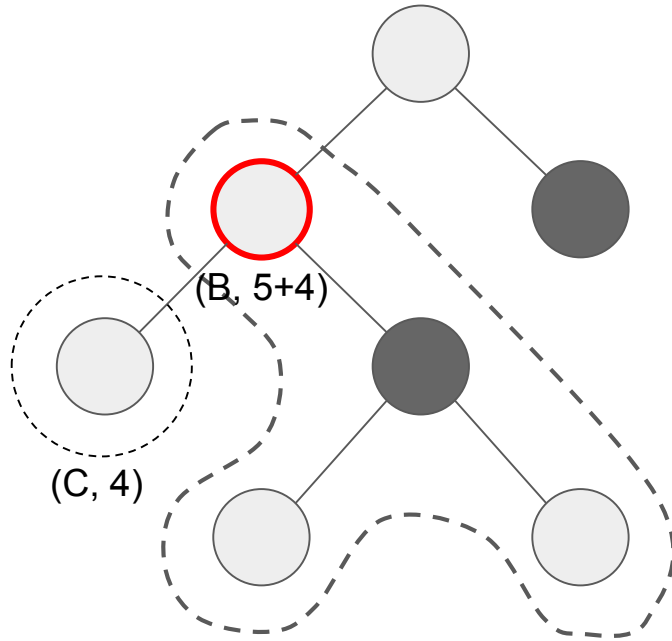
(B) Cost: 5



(D) Cost: 3

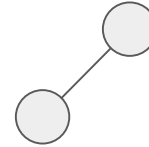


Input Tree



Sub Trees

(A) Cost: 2

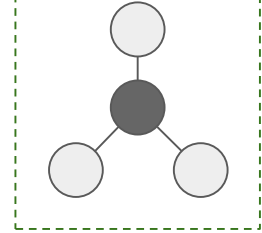


(C) Cost: 4



MATCH

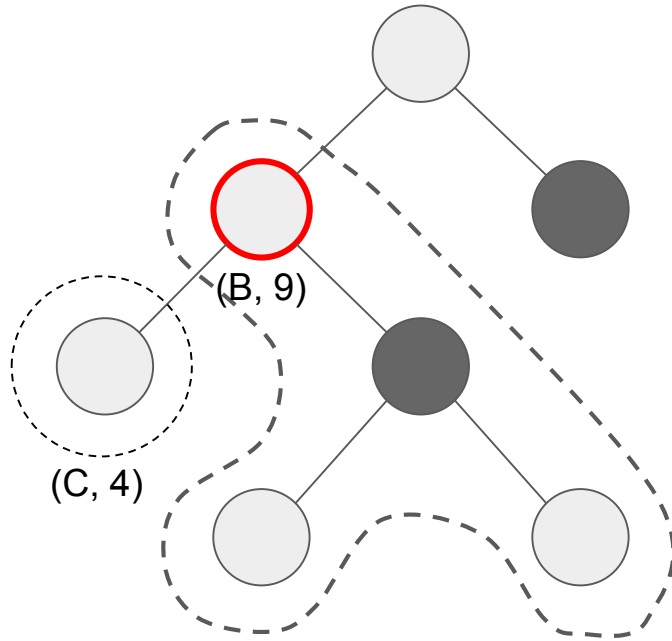
(B) Cost: 5



(D) Cost: 3

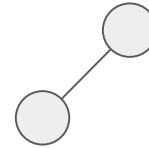


Input Tree



Sub Trees

(A) Cost: 2

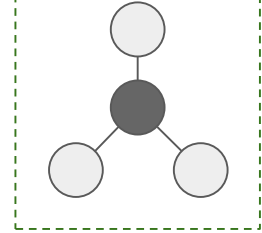


(C) Cost: 4



MATCH

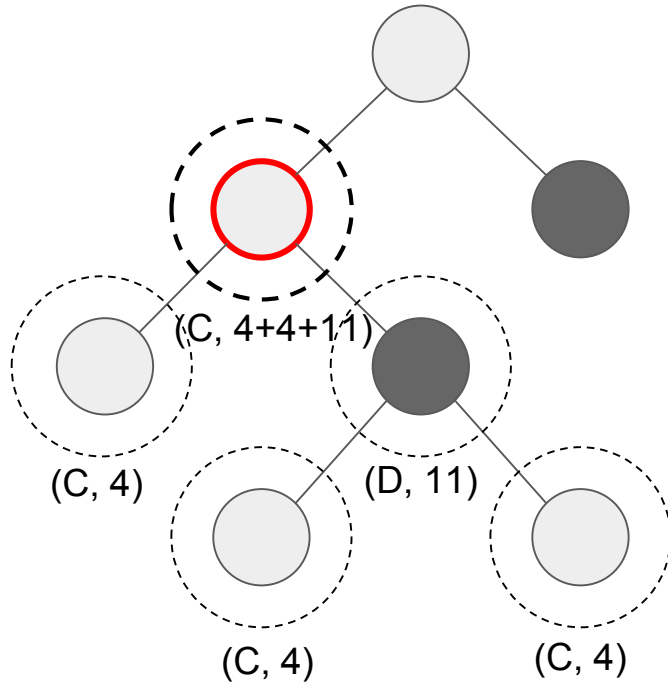
(B) Cost: 5



(D) Cost: 3

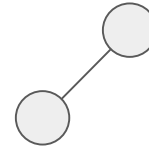


Input Tree

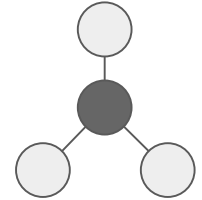


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

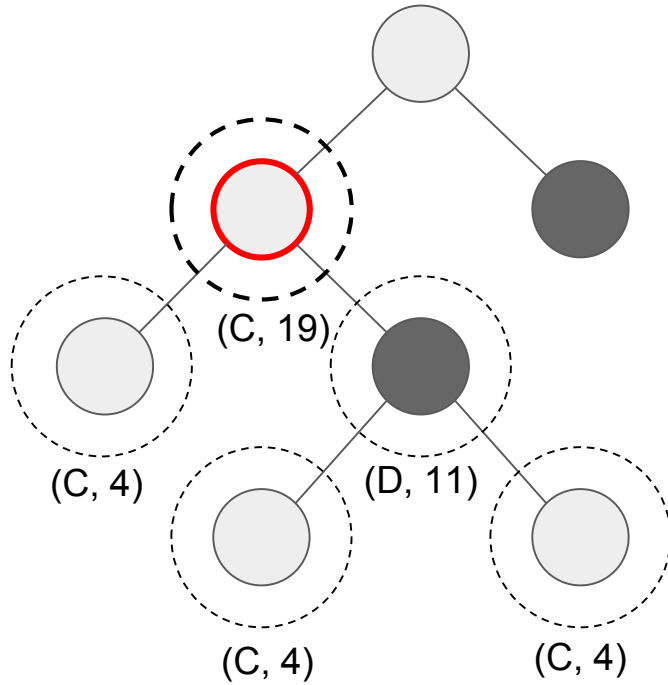
(C) Cost: 4



(D) Cost: 3

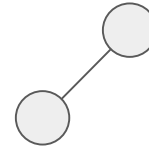


Input Tree

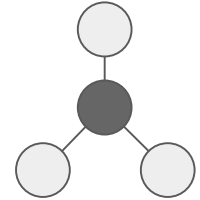


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

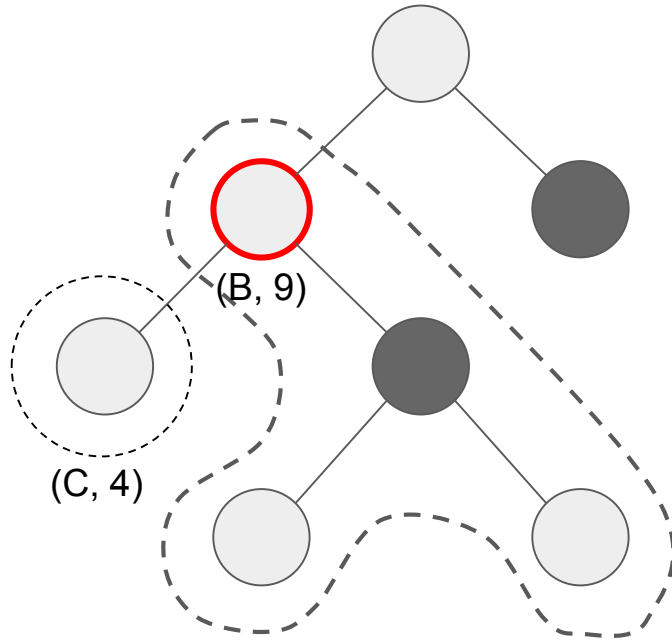
(C) Cost: 4



(D) Cost: 3

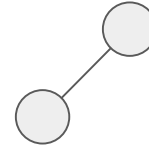


Input Tree



Sub Trees

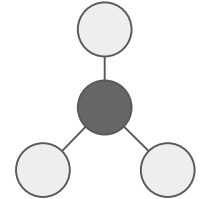
(A) Cost: 2



(C) Cost: 4



(B) Cost: 5

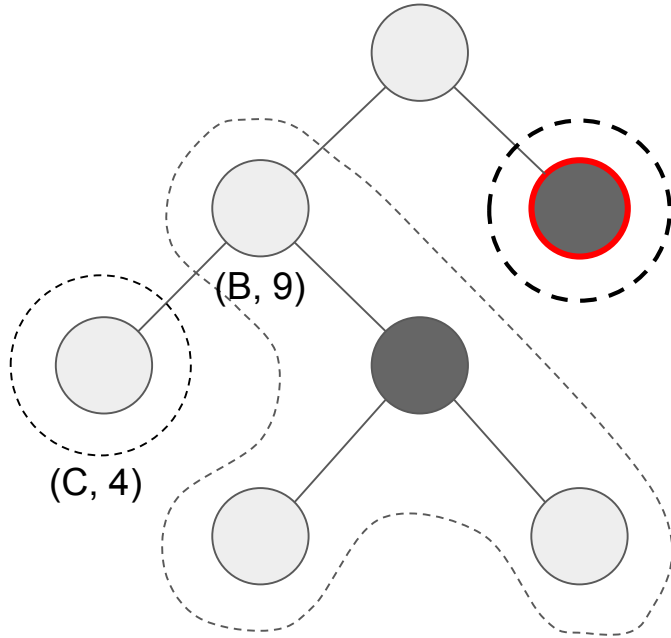


NO MATCH

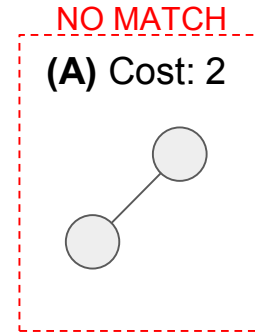
(D) Cost: 3



Input Tree



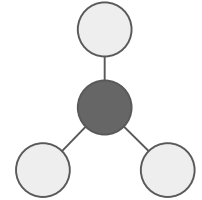
Sub Trees



(C) Cost: 4



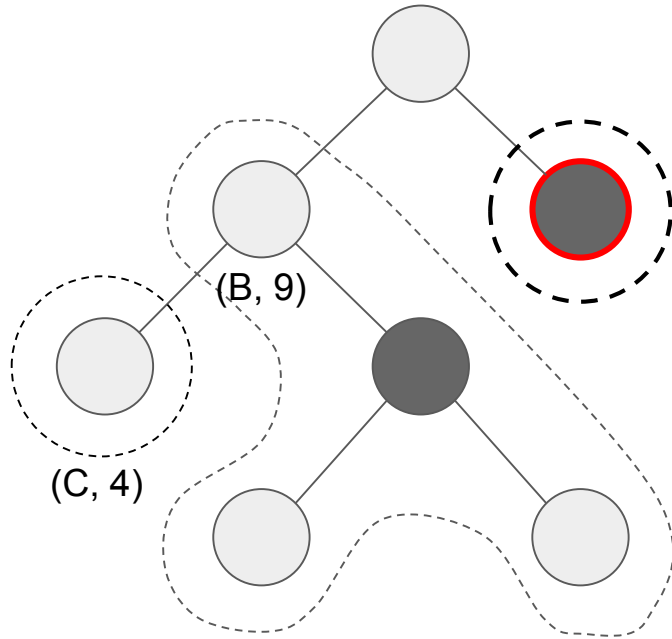
(B) Cost: 5



(D) Cost: 3

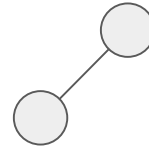


Input Tree



Sub Trees

(A) Cost: 2

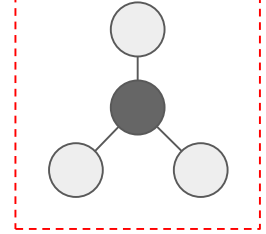


(C) Cost: 4



NO MATCH

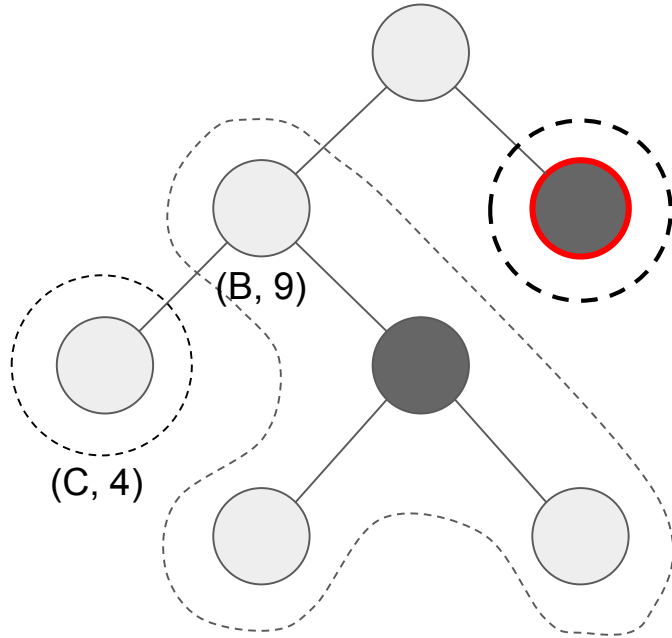
(B) Cost: 5



(D) Cost: 3

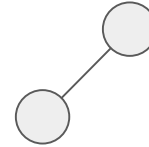


Input Tree

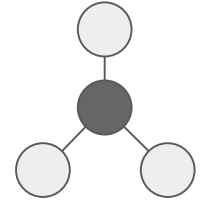


Sub Trees

(A) Cost: 2



(B) Cost: 5



NO MATCH

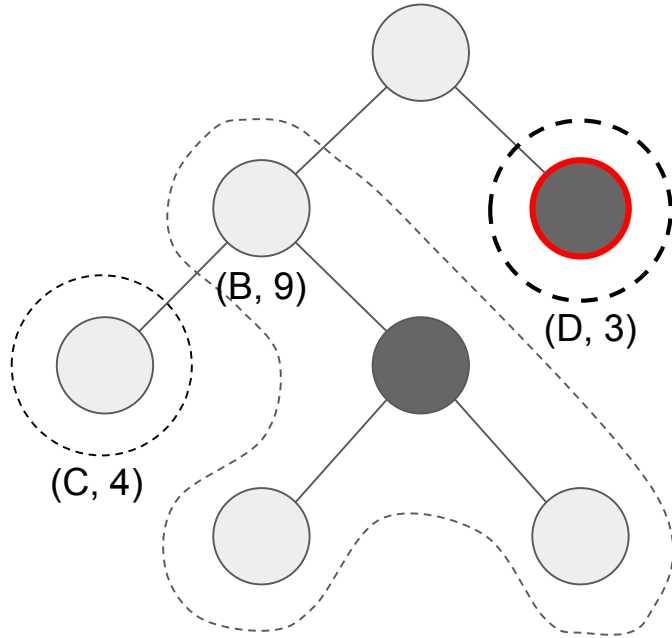
(C) Cost: 4



(D) Cost: 3

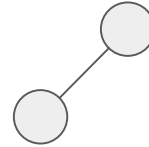


Input Tree

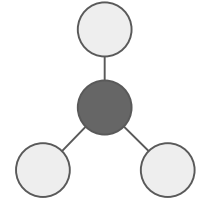


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

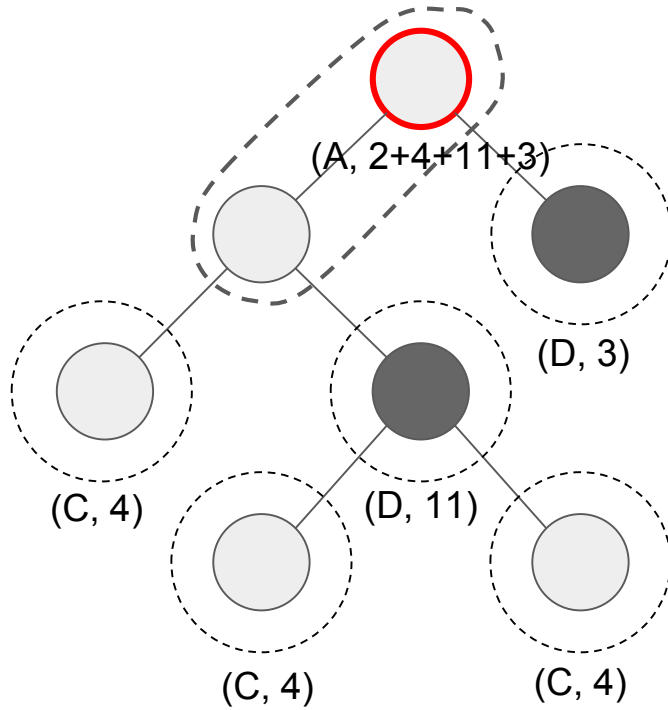
(C) Cost: 4



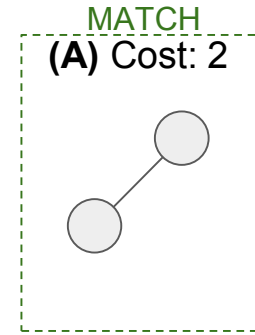
(D) Cost: 3



Input Tree



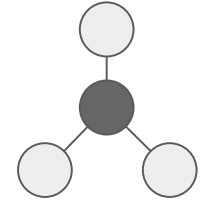
Sub Trees



(C) Cost: 4



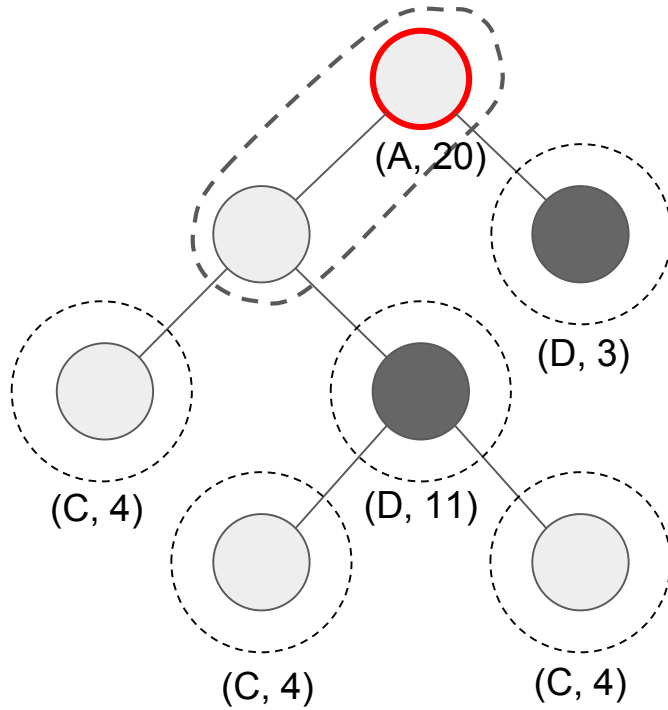
(B) Cost: 5



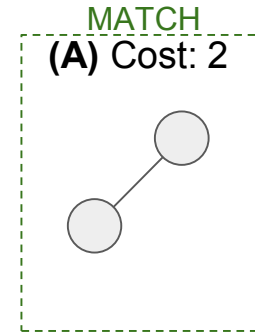
(D) Cost: 3



Input Tree



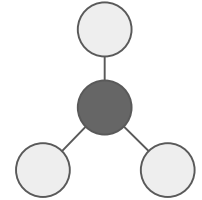
Sub Trees



(C) Cost: 4



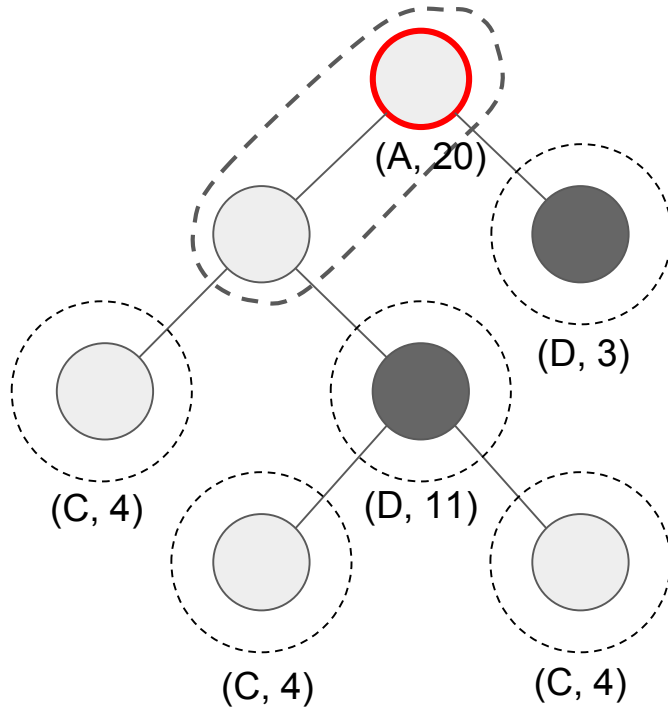
(B) Cost: 5



(D) Cost: 3

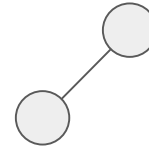


Input Tree



Sub Trees

(A) Cost: 2

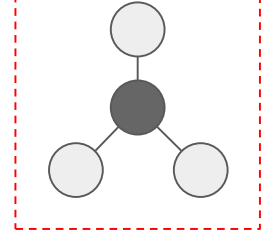


(C) Cost: 4



NO MATCH

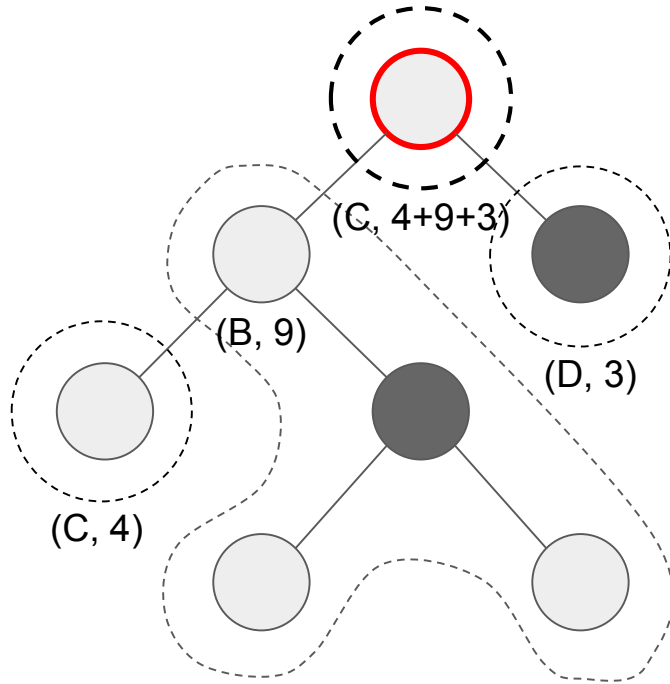
(B) Cost: 5



(D) Cost: 3

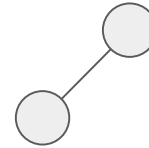


Input Tree

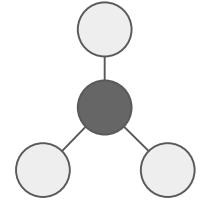


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

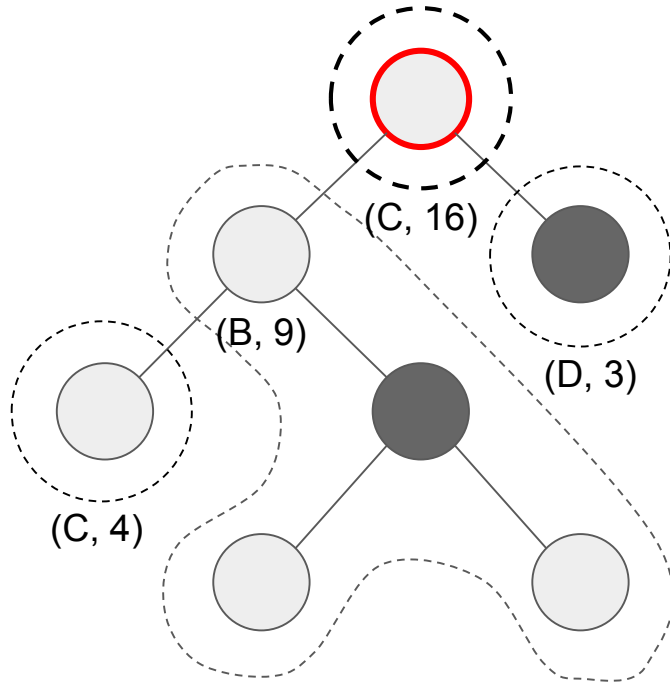
(C) Cost: 4



(D) Cost: 3

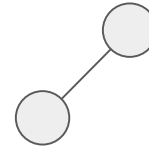


Input Tree

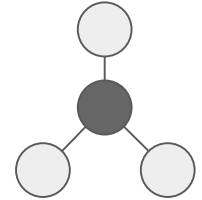


Sub Trees

(A) Cost: 2



(B) Cost: 5



MATCH

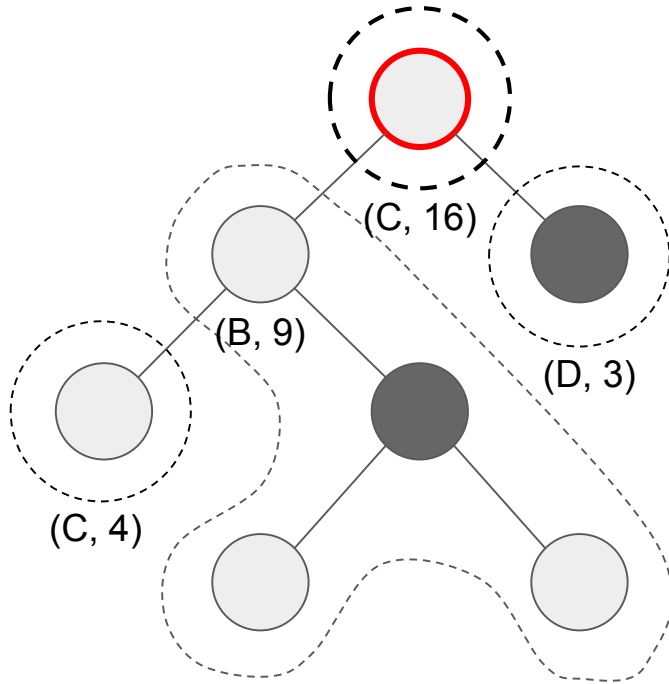
(C) Cost: 4



(D) Cost: 3

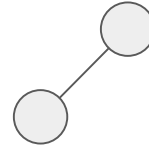


Input Tree

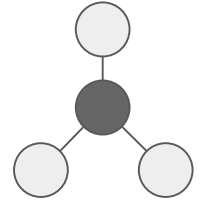


Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4

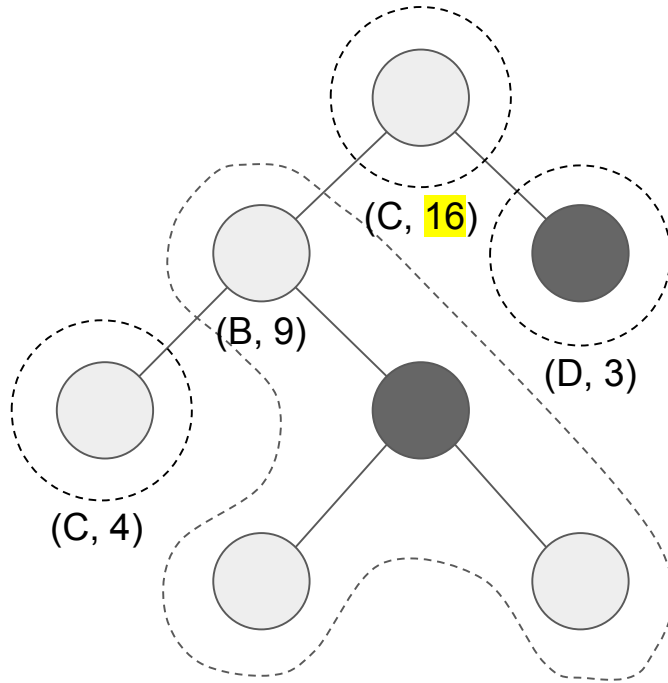


NO MATCH

(D) Cost: 3



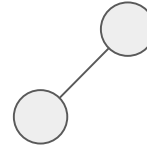
Input Tree



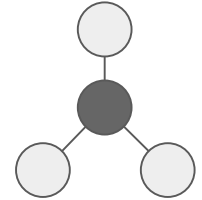
Optimum cost: 16

Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4



(D) Cost: 3

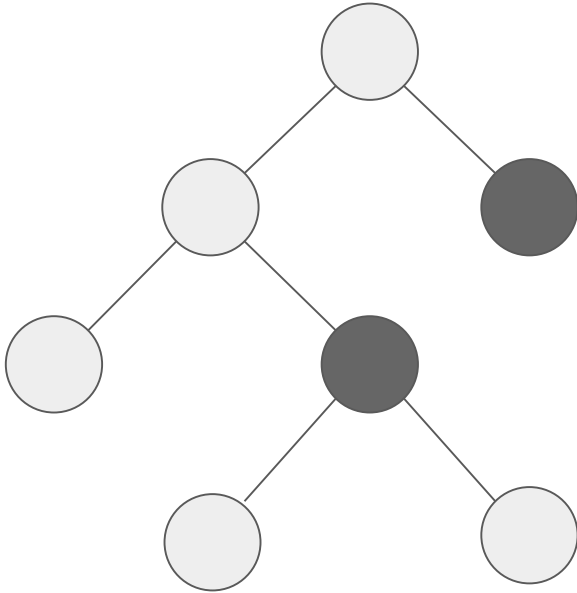


Questions

- Time complexity for an algorithm implementing tree tiling with dynamic programming?
- Result of applying Maximal Munch to this example=?

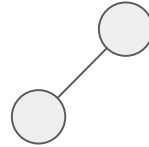
Using Maximal Munch

Input Tree

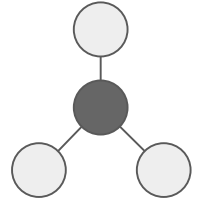


Sub Trees

(A) Cost: 2



(B) Cost: 5



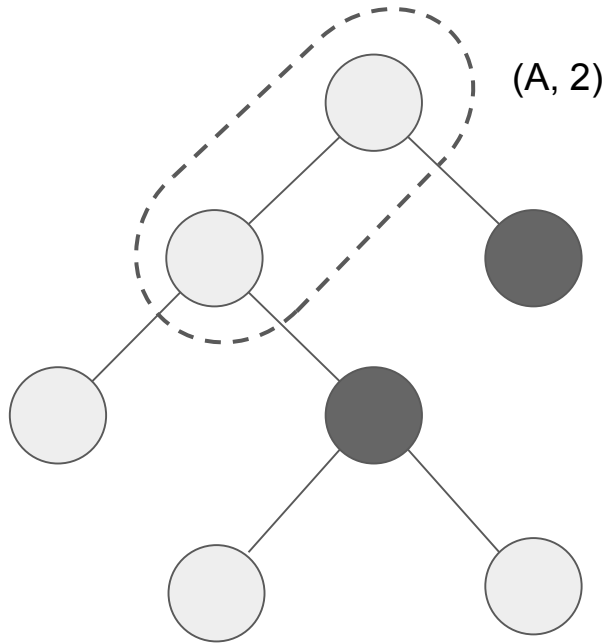
(C) Cost: 4



(D) Cost: 3

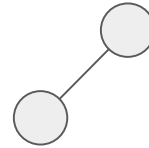


Input Tree

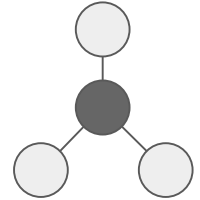


Sub Trees

(A) Cost: 2



(B) Cost: 5



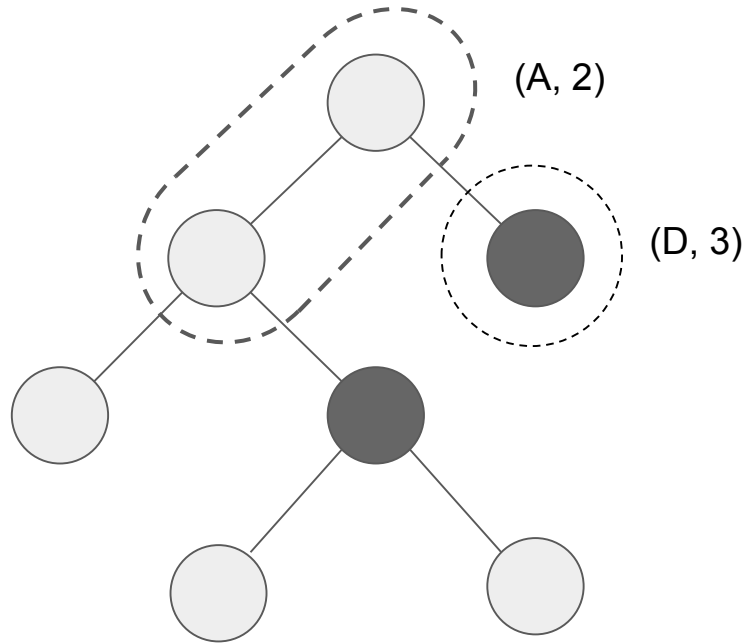
(C) Cost: 4



(D) Cost: 3

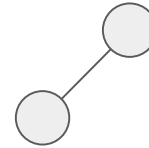


Input Tree

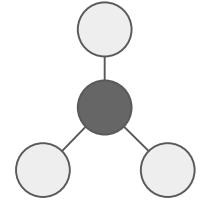


Sub Trees

(A) Cost: 2



(B) Cost: 5



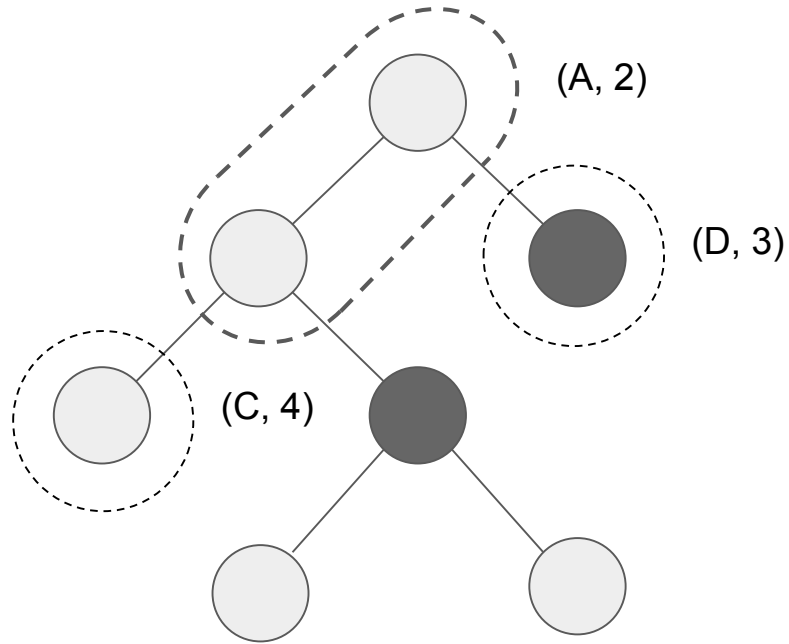
(C) Cost: 4



(D) Cost: 3

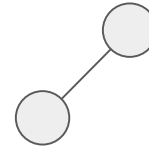


Input Tree

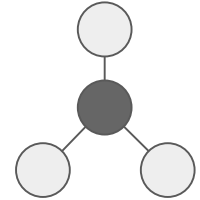


Sub Trees

(A) Cost: 2



(B) Cost: 5



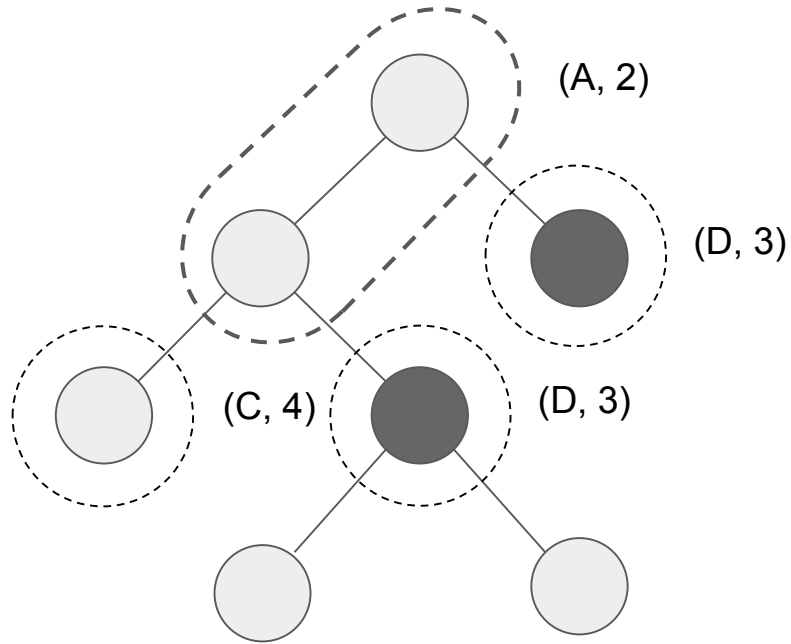
(C) Cost: 4



(D) Cost: 3

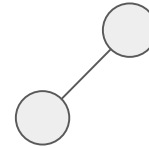


Input Tree

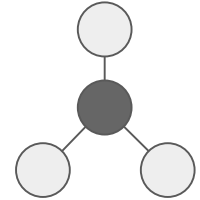


Sub Trees

(A) Cost: 2



(B) Cost: 5



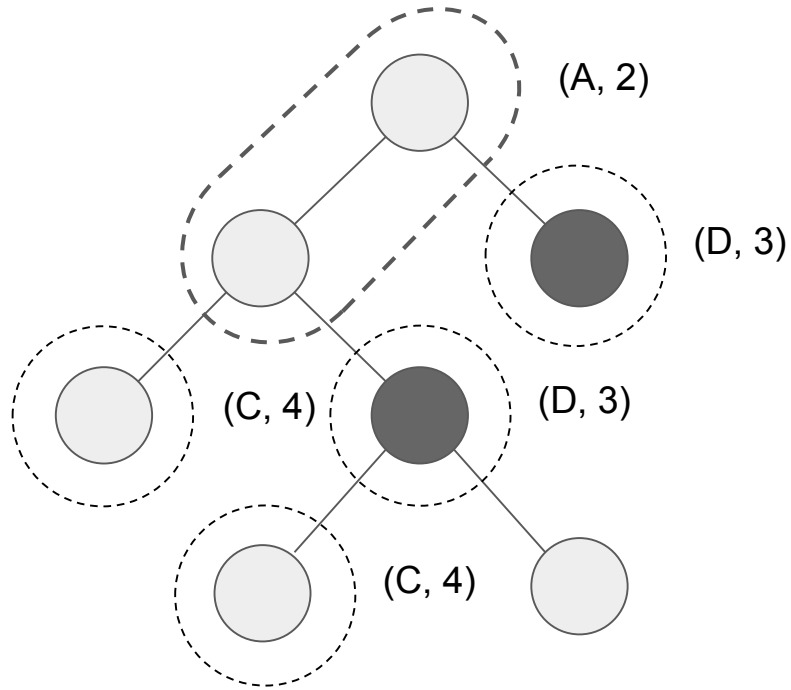
(C) Cost: 4



(D) Cost: 3

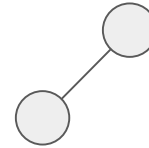


Input Tree

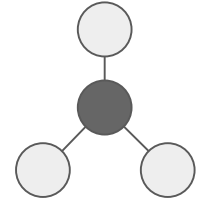


Sub Trees

(A) Cost: 2



(B) Cost: 5



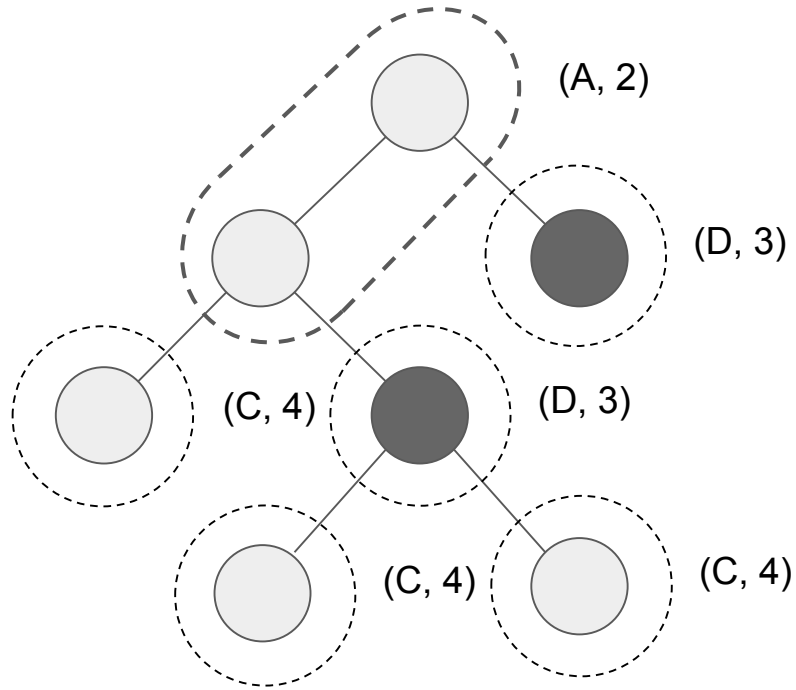
(C) Cost: 4



(D) Cost: 3

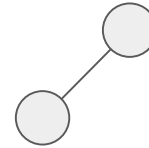


Input Tree

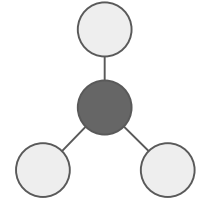


Sub Trees

(A) Cost: 2



(B) Cost: 5



(C) Cost: 4



(D) Cost: 3

