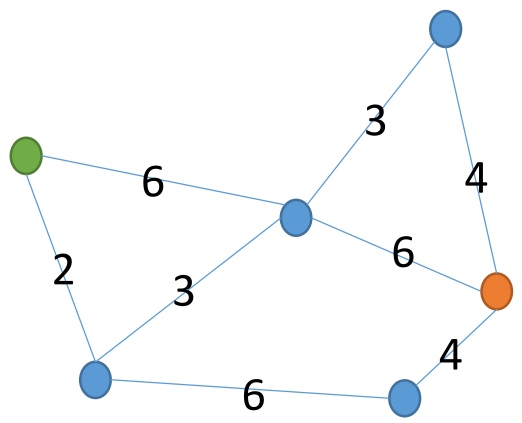
Exercise Set 1

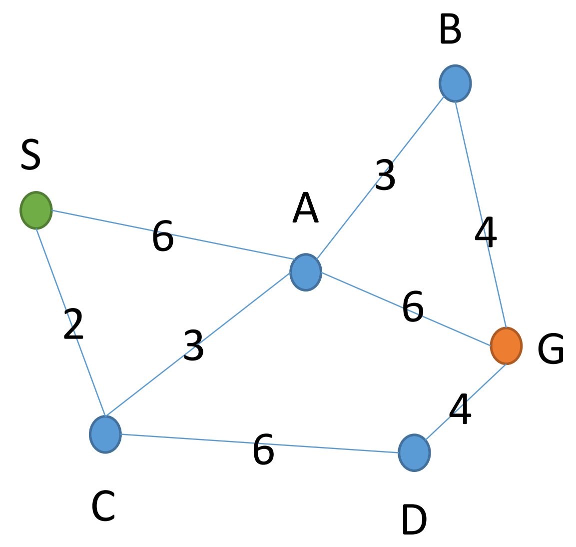
Pathfinding using A*

Examine the graph to the right. The green node is the origin, the red the goal.

1. Provide an optimistic and monotonic heuristic.
2. Perform Tree A* (notice that the edges are undirected – you can transition in both directions) to calculate the best path from origin to goal. Provide the frontier for each step and the node expanded, and remember to track all required information.



Node	Heuristic
S	4
A	2
B	2
C	4
D	2
G	0



The heuristic is the number of steps from goal, multiplied by the smallest edge cost (2).

Best Path: S,C,A,G.
Cost: 11

Step	Expanding	Frontier
0		$S^{<>}_{0+4=4}$
1	$S^{<>}_{0+4=4}$	$C^{<S>}_{2+4=6}, A^{<S>}_{6+2=8}$
2	$C^{<S>}_{2+4=6}$	$A^{<S,C>}_{5+2=7}, S^{<S,C>}_{4+4=8}, D^{<S,C>}_{8+2=10}$
3	$A^{<S,C>}_{5+2=7}$	$S^{<S,C>}_{4+4=8}, B^{<S,C,A>}_{8+2=10}, D^{<S,C>}_{8+2=10}, G^{<S,C,A>}_{11+0=11}$
4	$S^{<S,C>}_{4+4=8}$	$B^{<S,C,A>}_{8+2=10}, D^{<S,C>}_{8+2=10}, C^{<S,C,S>}_{6+4=10}, G^{<S,C,A>}_{11+0=11}, A^{<S,C,S>}_{10+2=12}$
5	$B^{<S,C,A>}_{8+2=10}$	$D^{<S,C>}_{8+2=10}, C^{<S,C,S>}_{6+4=10}, G^{<S,C,A>}_{11+0=11}, A^{<S,C,S>}_{10+2=12}$
6	$D^{<S,C>}_{8+2=10}$	$C^{<S,C,S>}_{6+4=10}, G^{<S,C,A>}_{11+0=11}, A^{<S,C,S>}_{10+2=12}$
7	$C^{<S,C,S>}_{6+4=10}$	$G^{<S,C,A>}_{11+0=11}, A^{<S,C,S>}_{10+2=12}, S^{<S,C,S,C>}_{8+4=12}, D^{<S,C,S,C>}_{14+2=16}$
8	$G^{<S,C,A>}_{11+0=11}$	

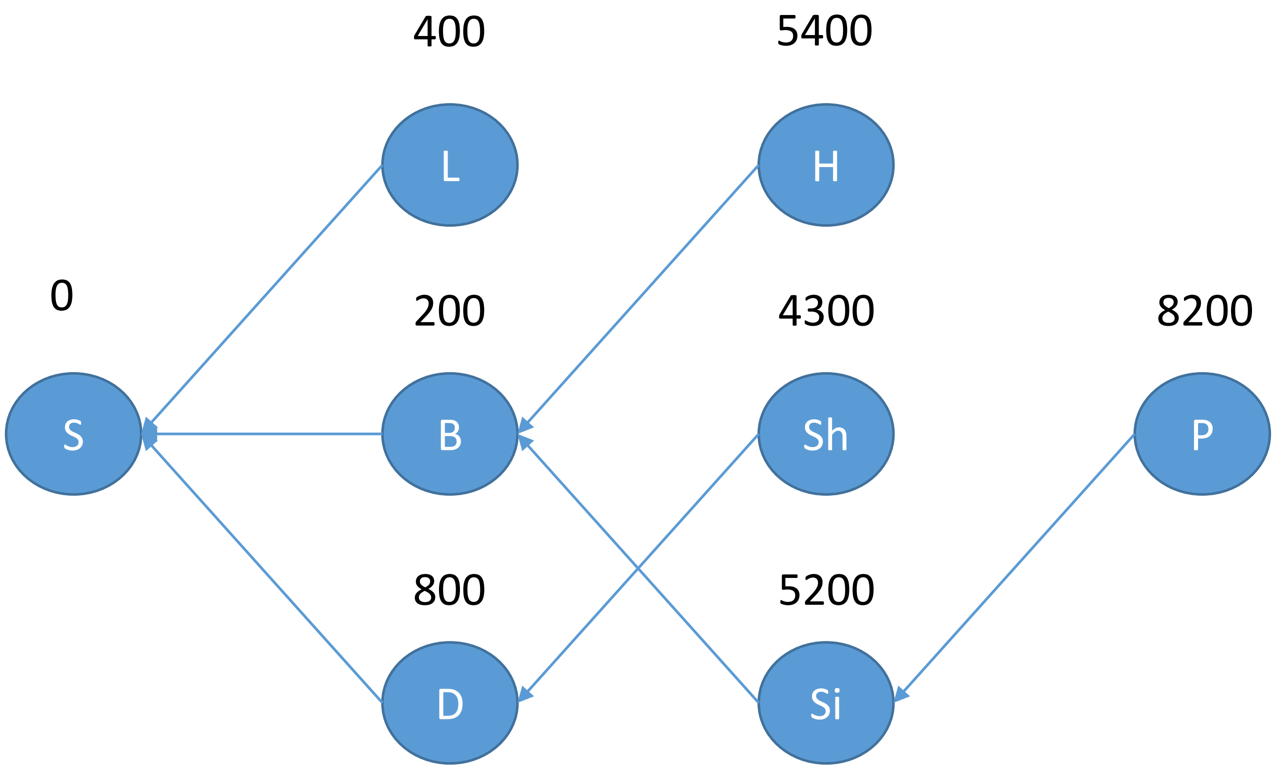
Pathfinding using Dynamic Programming

You need to travel from Stockholm to Palou. Ticket prices for possible legs are given on the table to the right. Find the cheapest route using the Dynamic Programming for Pathfinding algorithm.

1. Set up the layered space with edge costs. (How will you treat missing edges?)
2. Use the algorithm to solve the problem, showing all working.

From	To	Cost
Stockholm	London	400
Stockholm	Berlin	200
Stockholm	Dubai	800
London	Singapore	6000
London	Hong Kong	5000
London	Shanghai	4000
Berlin	Hong Kong	5000
Berlin	Singapore	5000
Dubai	Shanghai	3500
Dubai	Singapore	4500
Singapore	Palou	3000
Hong Kong	Palou	3500
Shanghai	Palou	5000

Cheapest Route: Stockholm – Berlin – Singapore – Palou.
Working not shown.



Minimax & Alpha-Beta Pruning

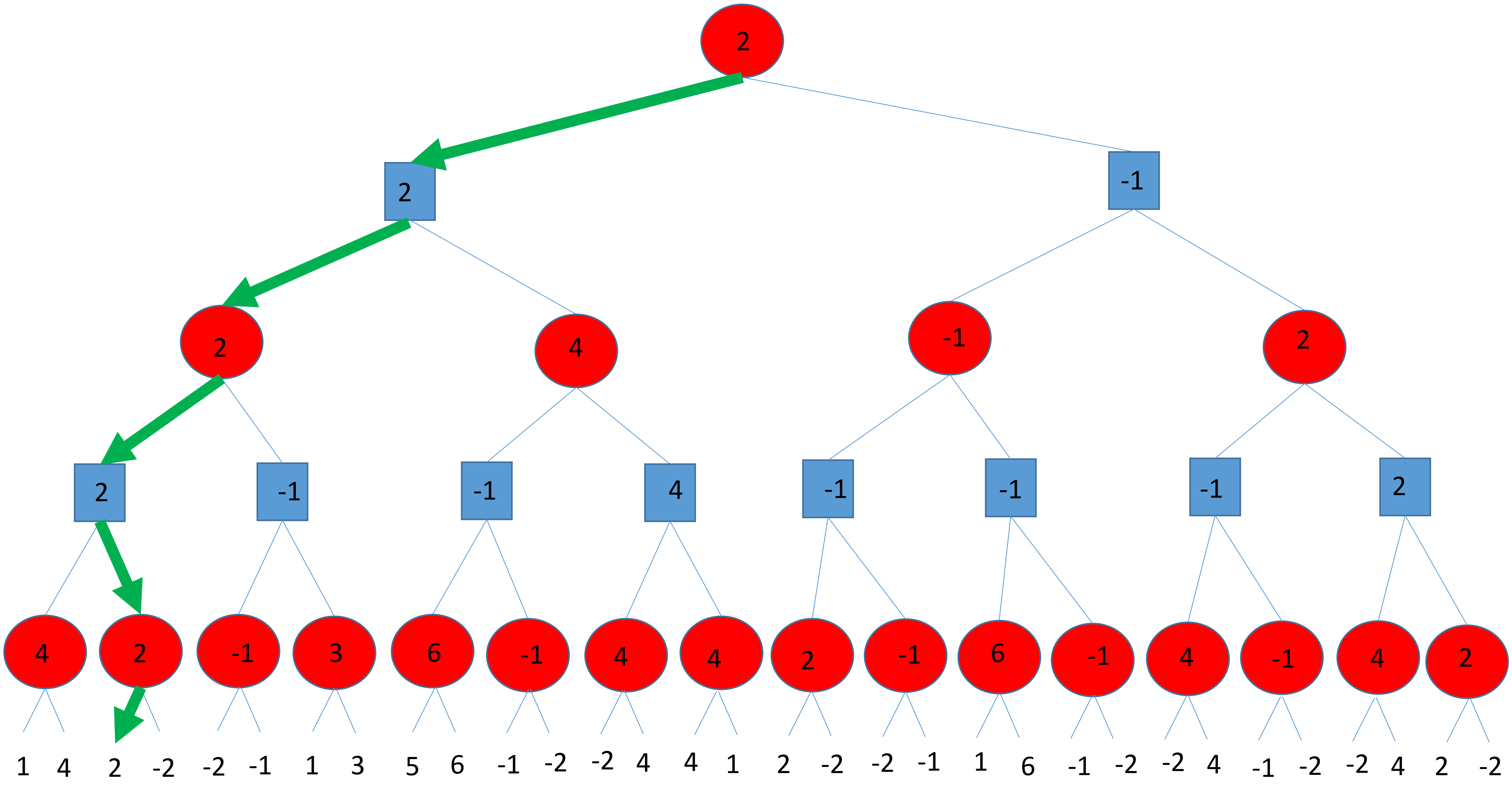
Examine the graphs on the next two pages.

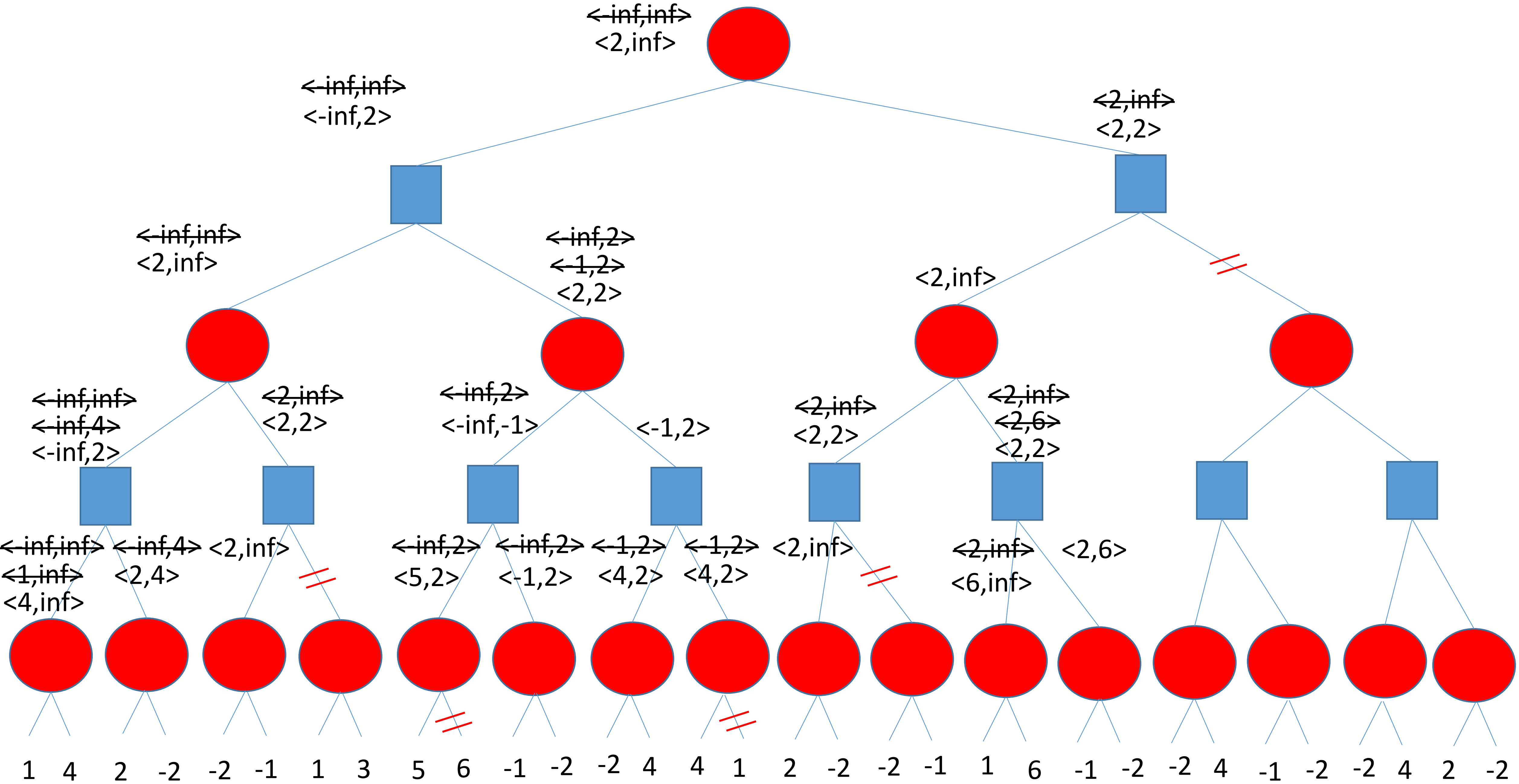
1. **On the first, perform the Minimax algorithm to work out the branch that will be played if both players play optimally.**
2. **On the second, perform alpha-beta pruning to find <u>the expected outcome of the game</u>.**

In both cases, assume the cut off depth if that of the last layer of nodes. At those nodes you should perform a heuristic evalation of the state of the game. For these exercises, this means simply making up a number! However:

   1. **Make up different estimates for each state (this avoids the possibility of equally optimal moves/branches).**
   2. **For the alpha-beta question, make up estimates that will ensure that at least some pruning occurs.**

This question has been altered where it is underlined.


2
2
-1
2
4
-1
2
2
-1
-1
4
-1
-1
-1
2
4
2
-1
3
6
-1
4
4
2
-1
6
-1
4
-1
4
2
1 4 2 -2 -2 -1 1 3 5 6 -1 -2 -2 4 4 1 2 -2 -2 -1 1 6 -1 -2 -2 4 -1 -2 -2 4 2 -2


~~<-inf,inf>~~
<2,inf>
~~<-inf,inf>~~
<-inf,2>
~~<2,inf>~~
<2,2>
~~<-inf,inf>~~
<2,inf>
~~<-inf,2>~~
~~<-1,2>~~
<2,2>
<2,inf>
~~<-inf,inf>~~
~~<-inf,4>~~
<-inf,2>
~~<2,inf>~~
<2,2>
~~<-inf,2>~~
<-inf,-1>
<-1,2>
~~<2,inf>~~
<2,2>
~~<2,inf>~~
~~<2,6>~~
<2,2>
~~<-inf,inf>~~
~~<1,inf>~~
<4,inf>
~~<-inf,4>~~
<2,4>
<2,inf>
~~<-inf,2>~~
<5,2>
~~<-inf,2>~~
<-1,2>
~~<-1,2>~~
<4,2>
~~<-1,2>~~
<4,2>
<2,inf>
~~<2,inf>~~
<6,inf>
<2,6>
1 4 2 -2 -2 -1 1 3 5 6 -1 -2 -2 4 4 1 2 -2 -2 -1 1 6 -1 -2 -2 4 -1 -2 -2 4 2 -2