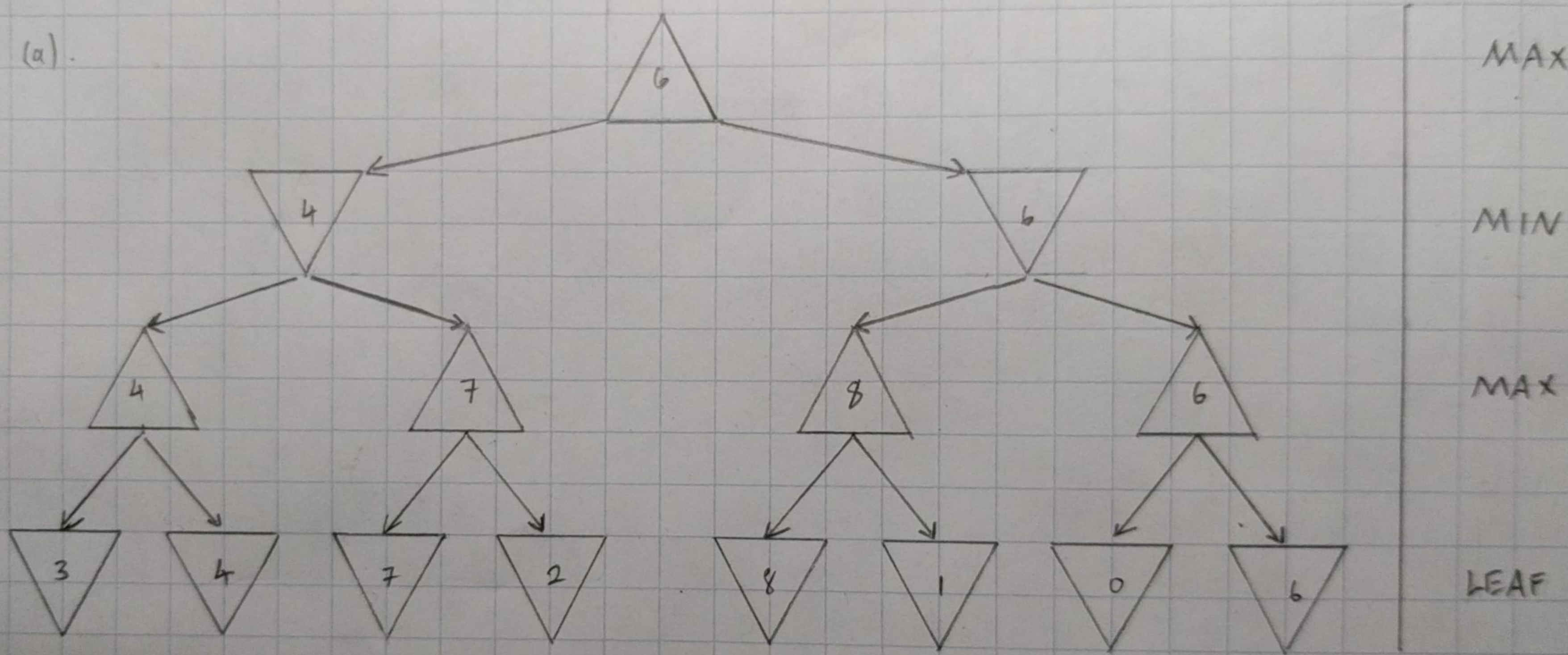
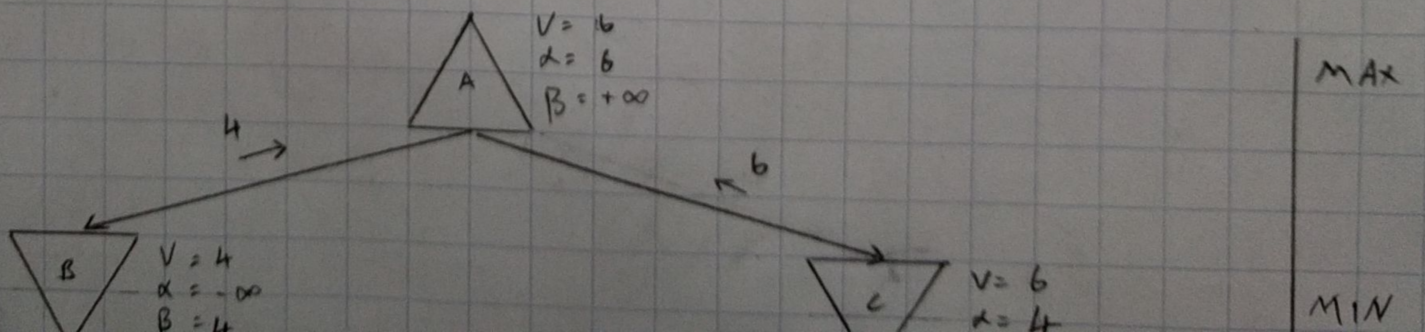
CS540 - HW5
Rouman Guha

Due: 03/12/18

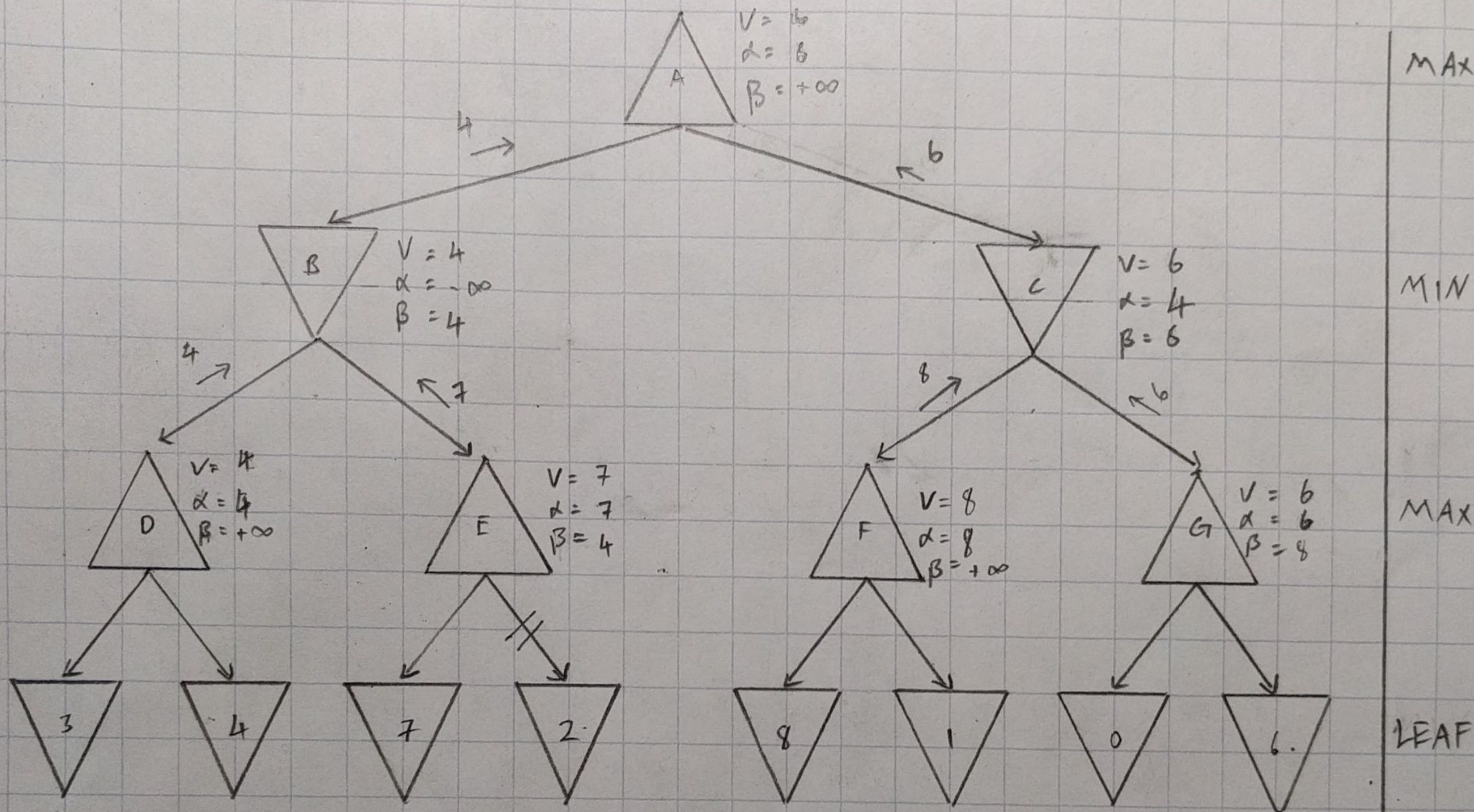
1. (a).



(b) α: best already-explored option along path to the root for maximizer.
β: best already-explored option along path to the root for minimizer.



(b) α: best already-explored option along path to the root for maximizer.
β: best already-explored option along path to the root for minimizer.




```
Max-Value (s, α, β) :
    if terminal (s) : return U(s);
    v = -∞;
    for c in next-states (s) :
        v' = Min-Value (c, α, β);
        if v' > v :  v = v';
        if v' ≥ β :  return v;
        if v' > α :  α = v';
    return v;
```

```
Min-Value (s, α, β) :
    if terminal (s) : return U(s);
    v = +∞;
    for c in next-states (s) :
        v' = Max-Value (c, α, β);
        if v' < v :  v = v';
        if v' ≤ α :  return v;
        if v' < β :  β = v';
    return v;
```

(c) The node with value 2 is pruned, as shown on the diagram.

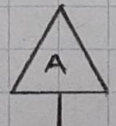
(d) Alpha-beta pruning is simply a more efficient implementation of Minimax, and in fact allows us to forego certain subtrees while still keeping the same final result.

(e). $P(\text{optimal choice}) = P(\text{heads}) + P(\text{tails}) \cdot P(\text{optimal choice} \mid \text{tails})$

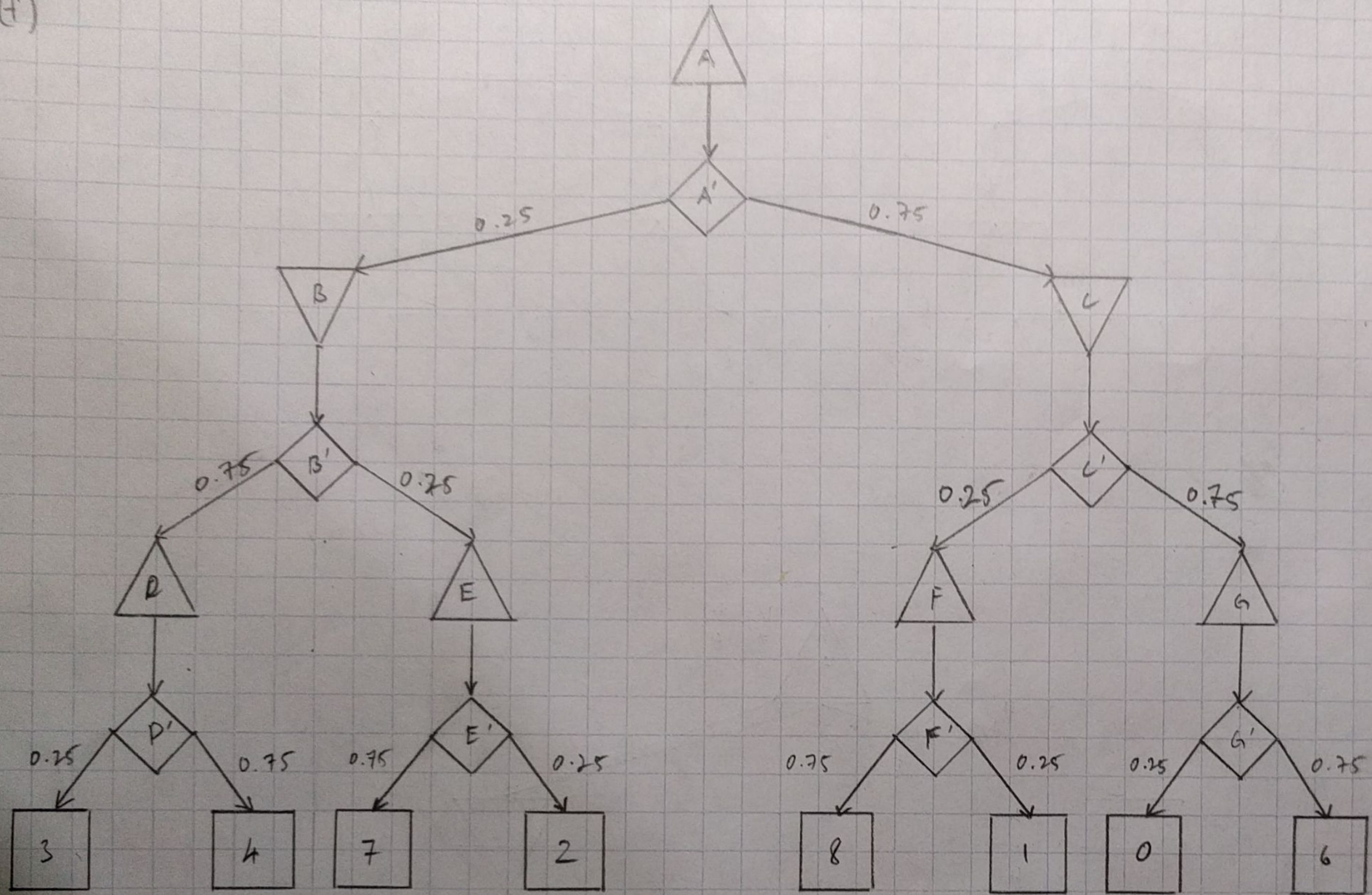
$= 0.5 + (0.5) \cdot (0.5)$

$= 0.75.$

(f)



(f)



(g).

$E(D) = 0.25(3) + 0.75(4)$
$= 3.75$

$E(E) = 0.75(7) + 0.25(2)$
$= 5.75$

$E(F) = 0.75(8) + 0.25(1)$
$= 6.25$

$E(G) = 0.75(6) + 0.25(0)$
$= 4.50$

$E(B) = 0.75(E(D)) + 0.25(E(E))$
$= 4.25$

$E(C) = 0.25(E(F)) + 0.75(E(G)).$
$= 4.94.$

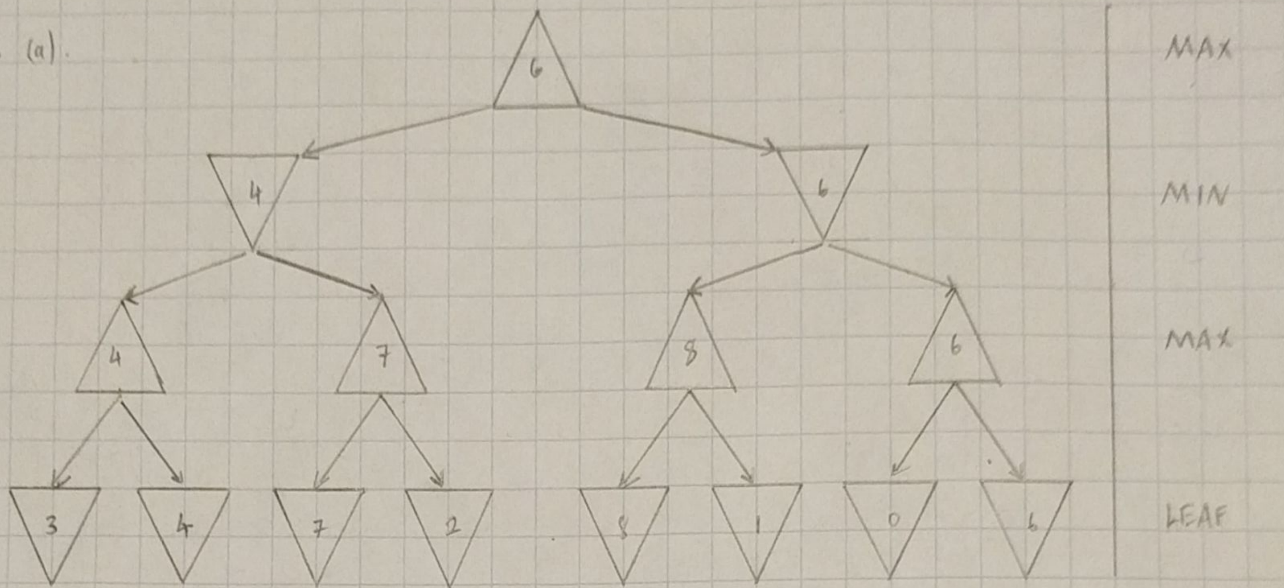
$E(A) = 0.25(E(B)) + 0.75(E(C))$
$= 4.77$

CS540 - HW5

Rouman Guha

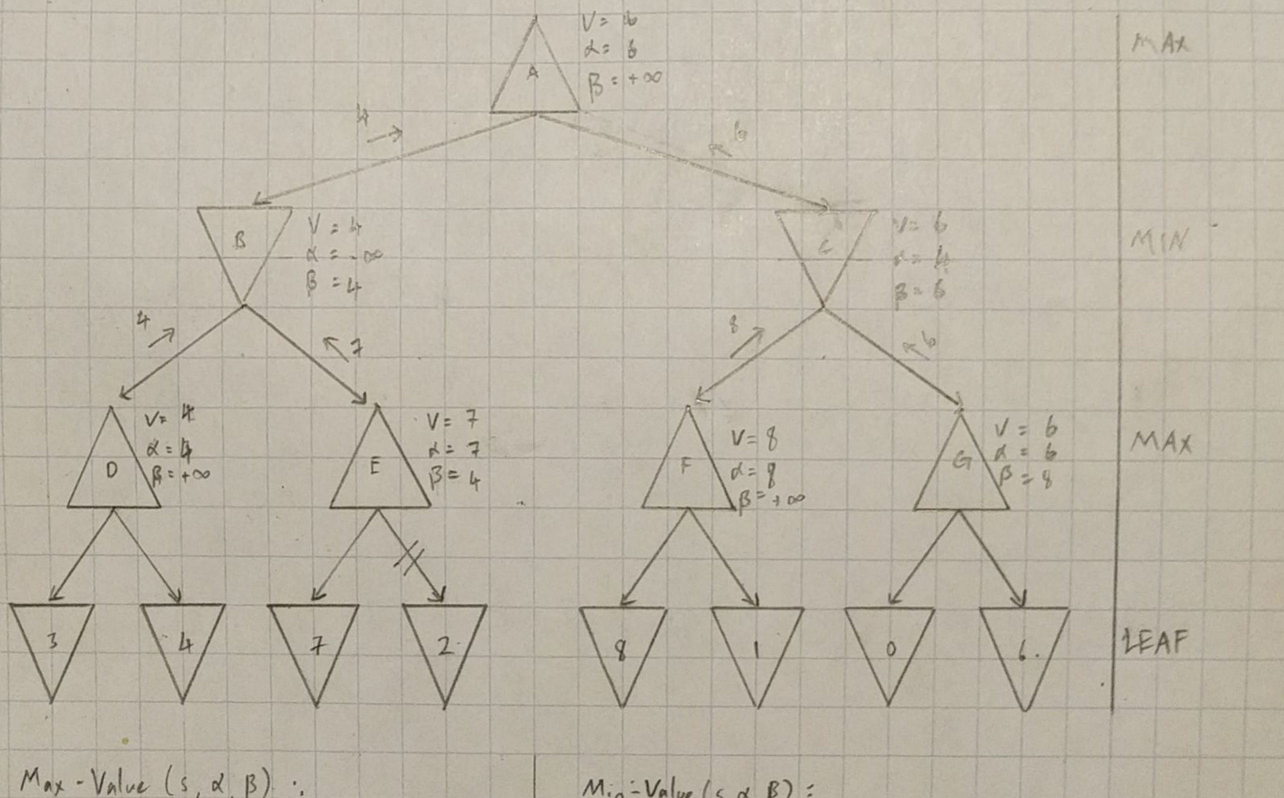
Due: 03/12/18

1. (a).



(b) α: best already-explored option along path to the root for maximizer.

β: best already-explored option along path to the root for minimizer.




```
Max-Value (s, α, β) :
    if terminal (s) : return U(s);
    v = -∞;
    for c in next-states (s) :
        v' = Min-Value (c, α, β);
        if v' > v : v = v';
        if v' ≥ β : return v;
        if v' > α : α = v';
    return v;
```

```
Min-Value (s, α, β) :
    if terminal (s) : return U(s);
    v = +∞;
    for c in next-states (s) :
        v' = Max-Value (c, α, β);
        if v' < v ; v = v';
        if v' ≤ α : return v;
        if v' < β ; β = v';
    return v;
```

(c) The node with value 2 is pruned, as shown on the diagram.

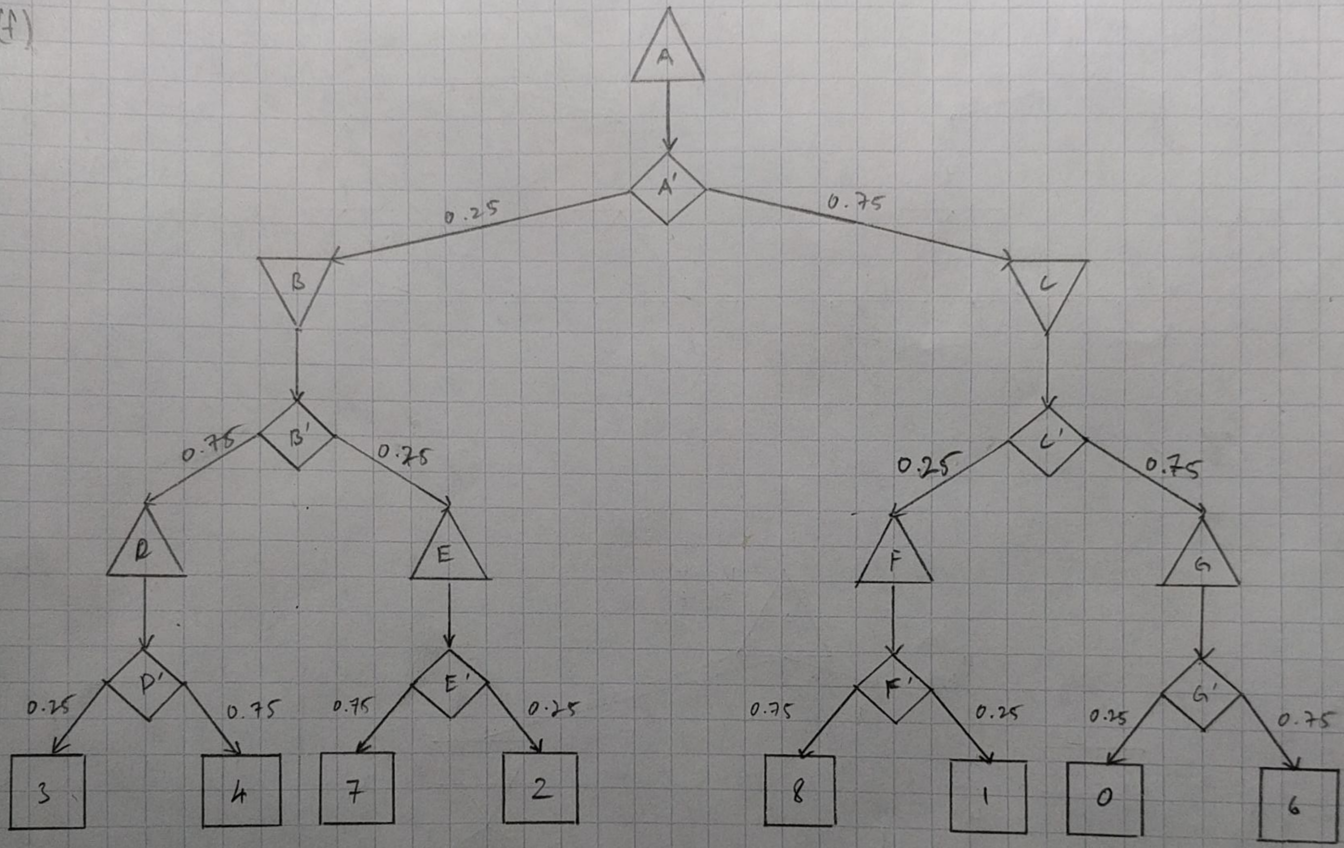
(d) Alpha-beta pruning is simply a more efficient implementation of Minimax, and in fact allows us to forego certain subtrees while still keeping the same final result.

(e). $P(\text{optimal choice}) = P(\text{heads}) + P(\text{tails}) \cdot P(\text{optimal choice} \mid \text{tails})$

$$= 0.5 + (0.5) \cdot (0.5)$$

$$= 0.75.$$

(f)



(g).

$$E(D) = 0.25(3) + 0.75(4) = 3.75$$

$$E(E) = 0.75(7) + 0.25(2) = 5.75$$

$$E(F) = 0.75(8) + 0.25(1) = 6.25$$

$$E(G) = 0.75(6) + 0.25(0) = 4.50$$

$$E(B) = 0.75(E(D)) + 0.25(E(E)) = 4.25$$

$$E(C) = 0.25(E(F)) + 0.75(E(G)). = 4.94.$$

$$E(A) = 0.25(E(B)) + 0.75(E(C)) = 4.77$$