

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## 2 MDD intersection

Perform the intersection of the MDDs given in Fig. 1. Detail all the steps, notably the reduction

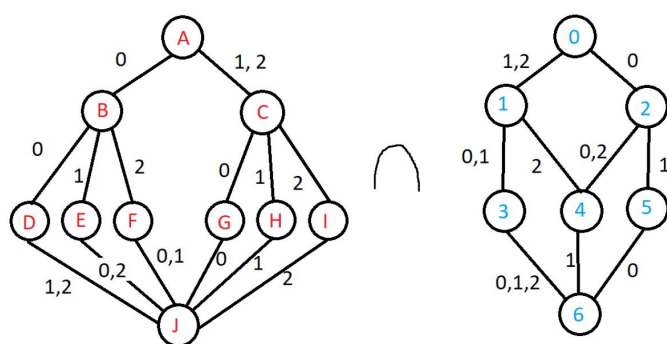


FIGURE 1 – Two MDDs

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

### 3 MDD union

Perform the union of the two MDDs given in Fig. 1.  
Explain your method and detail the steps.



## 5 Independent Set

An independent set in  $G$  is a subset of pairwise non-adjacent vertices.  
We consider the following problem :

DATA : Let  $G = (V, E)$  be a graph and  $k$  an integer.

QUESTION : Is there an independent set  $S$  of  $G$  such that  $|S| > k$

Here is a possible algorithm : We start with  $S \leftarrow \emptyset$

1. We identify  $x$ , the node in  $G$  having the minimum degree.
2. We add  $x$  to  $S$  and we remove  $x$  and its neighbors from  $G$
3. We go to 1 until there is at least one node in  $G$ .

Prove that this algorithm computes an independent set  $S$

---

---

---

---

---

---

---

---

Give an example solved by this algorithm

---

---

---

---

---

---

---

---

Give another example for which the algorithm is not good

---

---

---

---

---

---

---

---

Try to identify graphs for which the algorithm will not be able to compute a solution

---

---

---

---

---

---

---

---

## 6 Minimum Swaps for Bracket Balancing

You are given a string of  $2N$  characters consisting of  $N$  `[` brackets and  $N$  `]` brackets. A string is considered balanced if it can be represented in the form `S2[S1]` where `S1` and `S2` are balanced strings. We can make an unbalanced string balanced by swapping adjacent characters. Calculate the minimum number of swaps necessary to make a string balanced.

Examples :

Input : `[] [] [] [`

Output : 2

First swap: Position 3 and 4

`[] [] [] [`

Second swap: Position 5 and 6

`[] [] [] []`

Input : `[[] []]`

Output : 0

The string is already balanced.

We can solve this problem by using greedy strategies. If the first  $X$  characters form a balanced string, we can neglect these characters and continue on. If we encounter a `]` before the required `[`, then we must start swapping elements to balance the string.

Propose a greedy algorithm for this problem.

---

---

---

---

---

---

Show that your greedy is optimal

---

---

---

---

---

---