Approximation Algorithm

An approximation algorithm is the way of dealing with NP-completeness for optimization problems.

The goal of approximation algorithm is to give solution closer to optimal solution in polynomial time.

C → cost of solution given by Approximation Alg.

C^* → cost of optimal solution

$\rho(n)$ → Approximation Ratio

n → input size

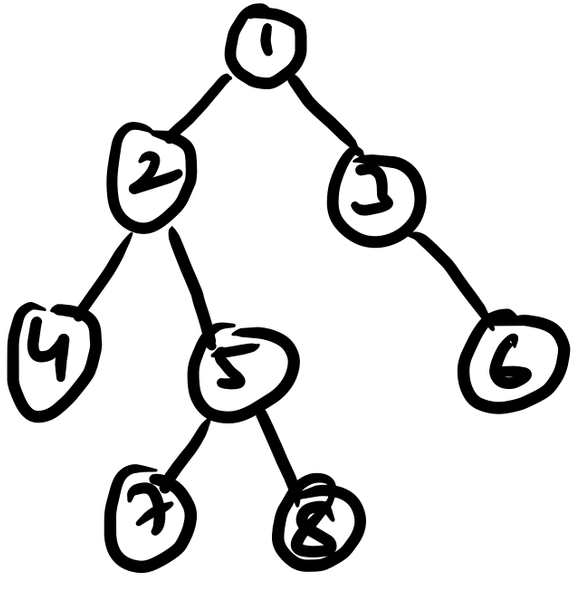
① Maximization: $\frac{C^*}{C} \leq \rho(n)$

② Minimization: $\frac{C}{C^*} \leq \rho(n)$: $\rho(n) \geq 1$

Vertex Cover Problem

A vertex cover of a graph G is a subset of vertices which cover every edges.

So, vertex cover problem is the minimum size vertex cover.



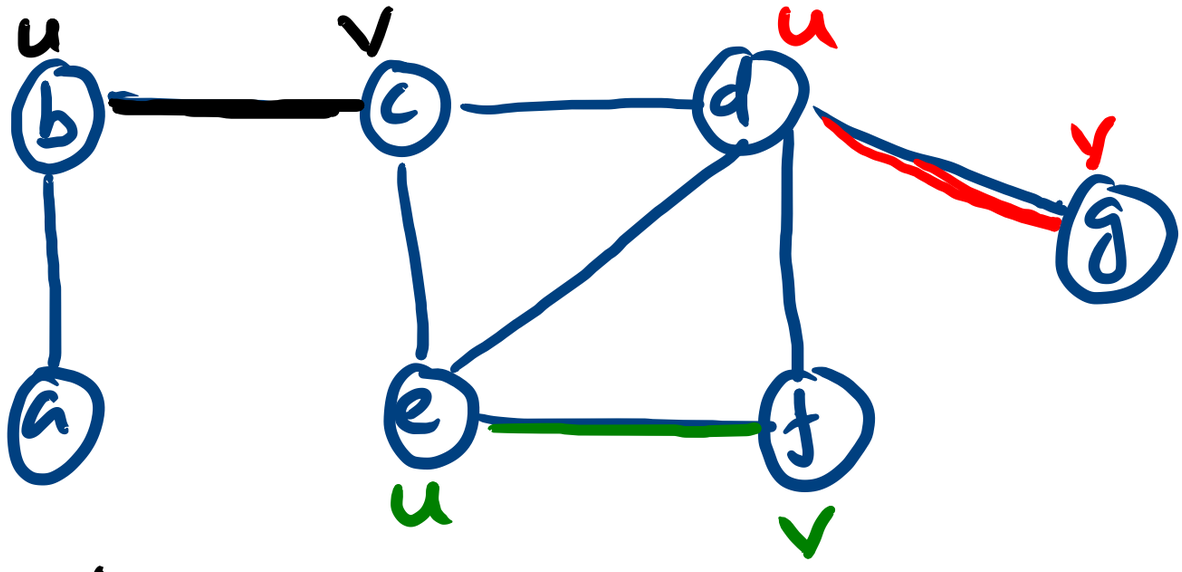
vertex cover = {1, 3, 2, 5}

= {5, 6, 2, 1}

= {2, 3, 5} ← min

Approximation – Vertex Cover (G)

1. $C \leftarrow \phi$
2. $E' \leftarrow E(G)$
3. while $E' \neq \phi$
   do let (u,v) an arbitrary edge in E'
4. $C \leftarrow C \cup \{u,v\}$
5. Remove from E' every edge incident either on u or v.
6. return C.



$E' = \{(a,b), (b,c), (c,d), (c,e), (d,e), (d,f), (e,f), (d,g)\}$

$C = \phi$

$= \phi \cup \{b,c\}$

$= \{b,c\}$

$= \{b,c\} \cup \{e,f\}$

$= \{b,c,e,f\}$

$C = \{b,c,e,f\} \cup \{d,g\}$

$= \{b,c,e,f,d,g\}$ Alle

↑ Approximation alg. $\Rightarrow C$

↓

6 Vertices ✓

but the optimal Solution for above graph G is

C^* $\{b, d, e\}$ i.e 3 vertices.

$\therefore$ Minimization $= \frac{C}{C^*}$

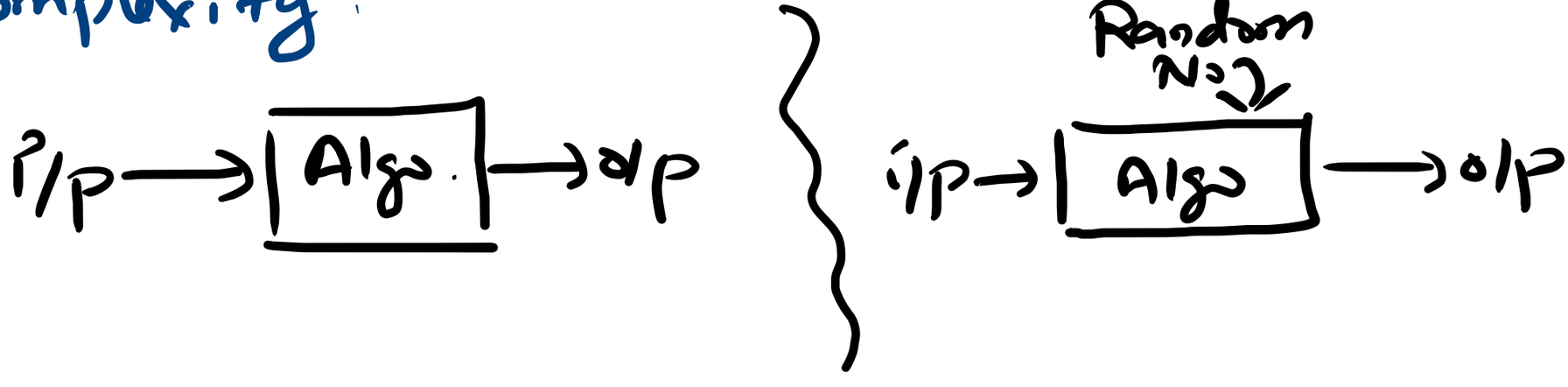
$= \frac{6}{3} = 2 > 1$

$\rho(n)$ ✓

Randomized Algorithm

An algorithm that uses random number to decide what to do next anywhere in its logic called the Randomized Algorithm.

- this algorithm is used to reduce space and time complexity.



① Las Vegas Algo

- output is always correct.

e.g. Randomized QuickSort

e.g.
```
AlgoSearch Repeat(A)
{
  for (i=0; i<=n-1; i++)
  {
    for (j=i+1; j<n; j++)
    {
      if (A[i]==A[j])
        return True
    }
  }
}
```

```
Algo Search Repeat_LV(A)
{
  while (True) do
    i = random() mod n+1
    j = random() mod n+1
  if ((i≠j) and (A[i]==A[j]))
    return True
}
```

② Monte Carlo Algo

- output may be incorrect

e.g. Randomized median

e.g.
```
AlgoSearch (A, K)
{
  for (i=0; i<=n-1; i++)
  {
    if (A[i]==K)
      return True
  }
}
```

```
Algo Monte Carlo_Search (A, K, x)
{
  for (i=0, i<=x; i++)
  {
    j = random() mod n+1
    if (A[j]==K)
      return True
  }
}
```