Algo - 2

23/10/2024

আমরা Algorithm কেন শিখি?
কোন situation এ কোনটা optimized solution সেটা জানার জন্য।

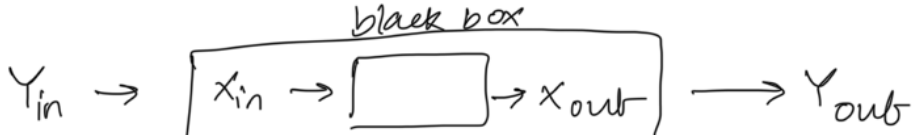
Fibonacci's rabbits ← Italian mathematician Leonardo of Pisa

→ এখানে যেকো Fibonacci series এর recurrence relation আসছে।

$$T(n) = T(n-1) + T(n-2) + \text{some constant}$$

Solution: $O(2^n)$ [Exponential time]
Another technique: Memoization $O(n)$ [Linear time]

Q. Why do we study computational complexity?
Garry and Johnson → computer & Interactibility
একটা problem কে আরেকটা established problem এ map করে we can infer how difficult/complex our problem actually is, does its solution exist or am I incompetent. Or how much efficient the solution can be.
When I reduce Y to X, the solution is using X



Assume $y \leq_P x$ then ⟶ y can be solved in polynomial time if x can be solved in polynomial time.

y is reducible to x in polynomial time

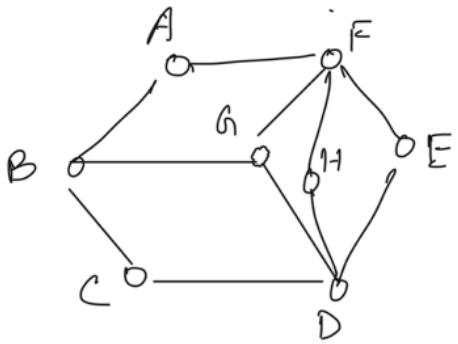
Contrapositive:

- X cannot be solved in polynomial time if Y cannot be solved in polynomial time.

Decision problem: যার answer Yes/No

Q. optimized problem কে decision problem এ কিভাবে convert করব? Through some constraint — e.g. at least (maximum) or at most (minimum)

→ hardness বোঝার সময়ে, optimized problem এর output এর উপর constraint দিয়েই we can make it a decision problem. Yes/No তে নিয়ে আসলে আমাদের জন্য handle করা easier.



Independent set of vertices
[কেউ কারো adjacent না, pairwise নেওয়া যাবেনা]

$S = \{A, E, \dots\}$

Q. Make two independent sets from this graph

maximum

maximal

[cannot be subset of any other independent set] [no superset exists]

[its cardinality should be greater/equal to any other independent set]

- independent set is not unique.

$S_1 = \{A, C, E, G, H\}$ → Maximal & Maximum

another independent set যোগ করা কিন্তু add করতে পারবোনা:

$S_2 = \{B, D, F\}$ → Maximal

maximum ই is also always maximal. But- there can be multiple maximal, and not all of them will be maximum.

Induced subgraph: যেই vertex গুলা নিয়ে আমরা subset যেগুলা connecting edge গুলো থাকবে।

∴ যেকোনো independent set এর induced subgraph is always a null graph.

Optimization problem: Give me the maximum independent set.

Decision problem: Can you give me an independent set of cardinality at least 5?

Vertex Cover: প্রতিটা edge থেকে একটি vertex নেওয়া মানে যেই edge কে cover করা

হয়েছে।

Now, optimization problem: সবচেয়ে কম vertex দিয়ে I want to cover all the edges.

{B, D, F}

Decision problem: can we find a vertex cover of at most size 3?

মুলত্ব language ↓: given a graph G and integer k, does G have a vertex cover set of size at most k?

Lemma: Let, $G = \langle V, E \rangle$ be a graph. Then a set $S \subseteq V$ is an independent set of G <u>if and only if</u> $V - S$ is a vertex cover of G.

[দুই side থেকেই prove করতে হবে]

→ prove this.

28 October, 2024

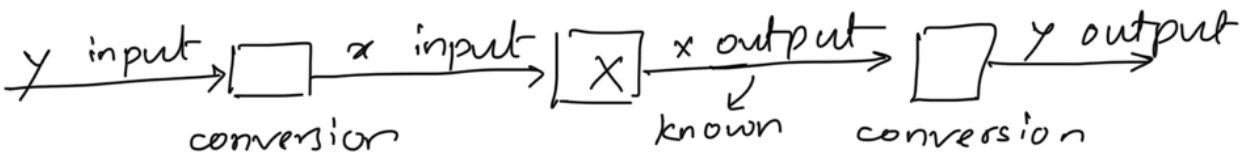
<u>Our goal</u>: There is a problem whose solution is not known. We want to prove that it is not less hard than another known problem y.

OR, x is at least as hard as y. [∴ y এর solution নাই মানে x এর ও solution নাই]

$x \longrightarrow y$

$x \longrightarrow y$

[x is reducible to y in polynomial time]



Optimization problem and decision problem এর hardness same but communicate করার জন্য decision problem more comfortable (Just Yes/No). এজন্যই আমরা এদের we convert optimization problem to decision problem.

$IS \leq_p VC$ [Independent Set reducible to Vertex Cover in polynomial time]

∴ VC এর solution থাকলে IS এর ও থাকবে।

There is a graph G which has a vertex cover of at most k vertices. [minimization problem]

Then:
The graph G has an independent set of at least (n-k) vertices.

[K & n-k vice versa]

- To prove $IS \leq_p VC$:

→ Take an input instance of Independent Set: Given graph G and an integer k, whether G has an independent set of at least k?

→ G has an IS of size k iff G has a vertex cover of n-k.

cover of n-k.

→ Then ask BB (Black box) whether G has a vertex cover of size n-k. [We have BB]

→ If yes: G has IS of size k ∴ Answer → Yes

• To prove $VC \leq_p IS$ (এবার IS এর BB আছে)

→ Take an input instance of vertex Cover: Given graph G and an integer k, whether G has a vertex cover of at most k vertices?

→ G has a VC of size k iff G has an independent set of n-k.

→ Then ask BB whether G has an independent set of size n-k.

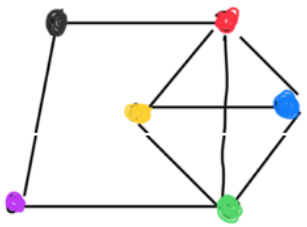
→ If yes: G has a VC of size k.

Answer: Yes

Clique (maximum subgraph)

From a graph, we have to find a set of vertices which make a complete graph.

Optimization problem: কোন clique সবচেয়ে বড়?



assignment: history of the word clique

· exam এ পাওয়া গেলে exercise করতে হবে।

[এই course এ আমরা যেসুলা graph নিয়ে কথা বলছি সবই simple graph: multi-edge নাই and self loop নাই]

complete graph এর complement : null graph

Q. What is complete graph?
→ যেই graph এ সবাই সবার neighbor.

Q. What is complement graph?

→ remove the edges between the vertices that exist in the original graph. And add the edges that did not exist.

Lemma: Graph G has an IS of size k ($|IS| = k$), iff $\bar{G}$ has a clique of size k

i.e. prove $IS \leq_p$ clique

[complement graph এর k size এর clique থাকলে graph এ k size এর IS থাকে]

Set Cover Problem

$U = \{\quad\}$

There are m subsets of U: $S_1, S_2, \ldots, S_m$.

- What is the minimum number of subsets which can be union-ed to get the original set?

(set cover and vertex cover কে relate করা যায়)

Decision problem: At most k টি union করলে U পাবো কিনা.

Lemma: G has $|VC| = k$ iff U has a set cover of K subsets.

↳ Hw for next class: how to map vertex cover problem to set cover problem.

3-set problem | Independent set

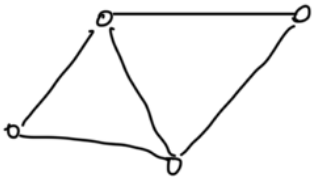
Boolean formula X in CNF with each clause in X is having three literal and output is yes/no depending X is satisfiable.

$X = \ldots \wedge \ldots \wedge \ldots \wedge \ldots$

30 October, 2024

How to map vertex cover problem to set cover problem

Graph



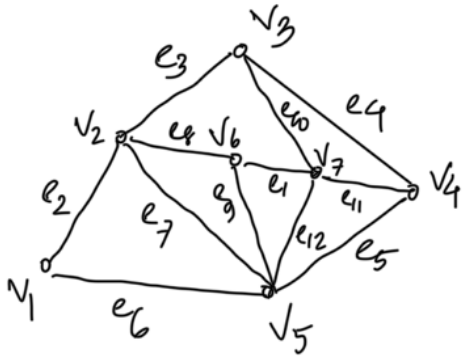
Vertex cover of at most k vertices

Set

U is a set.

subsets: $S_1, S_2, \ldots, S_m$

- Minimum কয়টি union করলে we will get U.



There are 7 vertices.

S_i = the set of edges which are incident on V_i

$S_1 = \{e_2, e_6\}$

$S_2 = \{e_2, e_7, e_8, e_3\}$

$S_3 = \{e_3, e_{10}, e_4\}$

$S_4 = \{e_4, e_{11}, e_5\}$

$S_5 = \{e_6, e_7, e_9, e_{12}, e_5\}$

$S_6 = \{e_8, e_9, e_1\}$

$S_7 = \{e_1, e_{10}, e_{11}, e_{12}\}$

Again, U = set of all edges

$= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}\}$

কোন S গুলা নিলে U পাবো?

$S_2 \cup S_7 \cup S_4 \cup S_5$ → size minimum হলেই হবে, solution unique না.

Again, vertex cover of the graph: $\{2, 4, 5, 7\}$

Lemma

Graph G has a vertex cover of size k iff set U has a set cover of size k OR

U can be covered with at most k of the sets $S_1, S_2, \ldots, S_m$.

necessity and sufficiency 2 দিকে prove করতে হবে।

3 Sat Problem (satisfiability problem)

Q. what is 3-sat problem?

[a boolean expression with some clause AND করা। And প্রতি clause এ 3টি করে literal আছে]

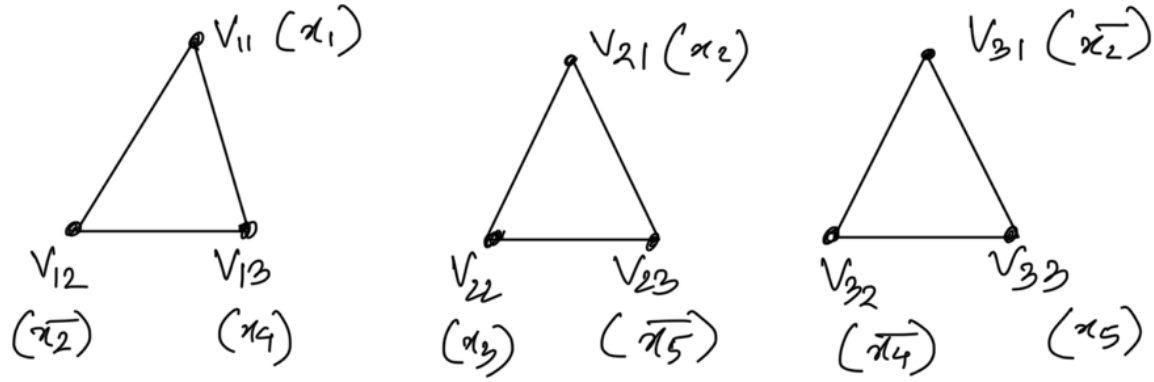
Q. what is satisfiability problem?

→ we have to reduce it to independent set problem.

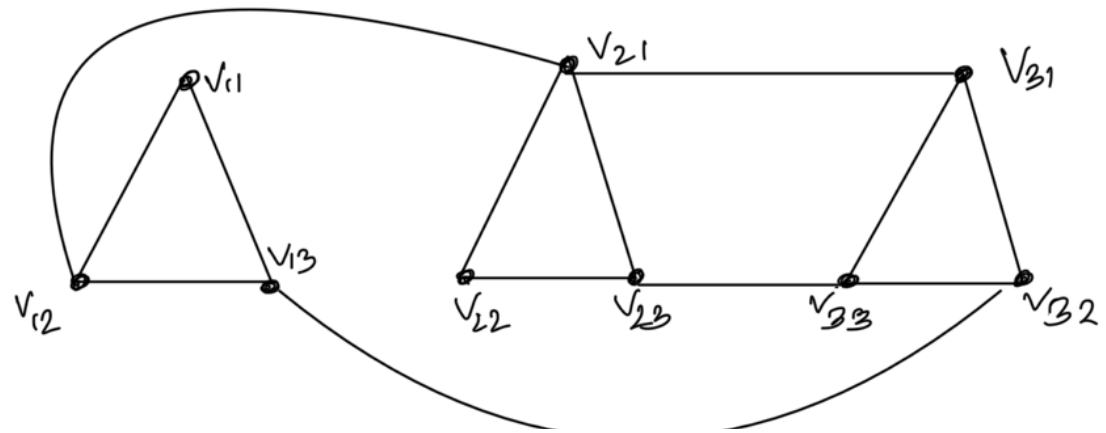
Lemma: x is satisfiable iff G has an Independent Set of size k → no. of clause

$$x = (x_1 + \overline{x_2} + x_4)(x_2 + x_3 + \overline{x_5})(\overline{x_2} + \overline{x_4} + x_5)$$

We will represent it as a graph. Each clause will be triangle. And the name of vertex will denote কোন clause এর কোন position এ আছে।



Then we will connect the complement variables (e.g. x_2 and $\overline{x_2}$)



এখন এটার independent set কী?

$\{V_{11}, V_{22}, V_{33}\}$

HW: Write the following proofs:

1. IS and VC
2. Clique and IS
3. VC and SC
4. 3-set and IS

Steps: ① state the problems and their corresponding decision problem.

② Prove the corresponding problem from both sides

③ Then reduce: - instance of A is given
- convert input instance of A into input instance of B
- then ask black box of B if that instance exists/yes/no
- if yes, then A. If no, then not A.

4 November, 2024 ♡

- tractable: polynomial time এ solvable
- intractable: polynomial time এ solvable না
- deterministic: যেকোনো step থেকে পরের step fixed
- non-deterministic: future এর অনেকগুলো possible options
- class P: polynomial time এ solvable problem
- class NP: Non-deterministic algorithm which

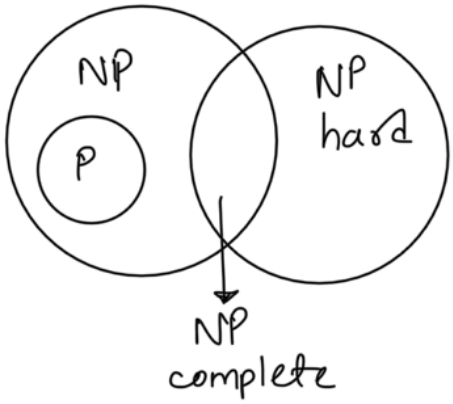
is solvable in polynomial time

For non-deterministic algorithms,
there is a guessing phase
and a verification phase (যেই guess করেছি, is it right?)
↳ can be done in polynomial time

#P class is part of NP.

NP hard: A problem x is NP hard if SAT problem can be reduced to x in polynomial time.
(SAT problem was the first NP problem) ← Cook's theorem

NP complete: An NP problem which is NP hard.



#এমন কিছু আছে করে যেটা NP না but NP hard

co-NP (Complement NP)

→ Complement of decision problem

Let, a decision problem:
Does a graph G have chromatic number 2?

Complement decision:

Does graph G have chromatic number anything other than 2?

একটার polynomial time a solution থাকলে কি complement টার ও polynomial time a solution থাকবে? We can't say for sure.

$$NP \neq CO\text{-}NP$$

Lemma: If $NP \neq CO\text{-}NP$ then $P \neq NP$

Contrapositive:

If $P = NP$ then $NP = CO\text{-}NP$

Let, $P = NP$

$x \in NP$

or, $x \in P$

or, $\bar{x} \in$ (?)

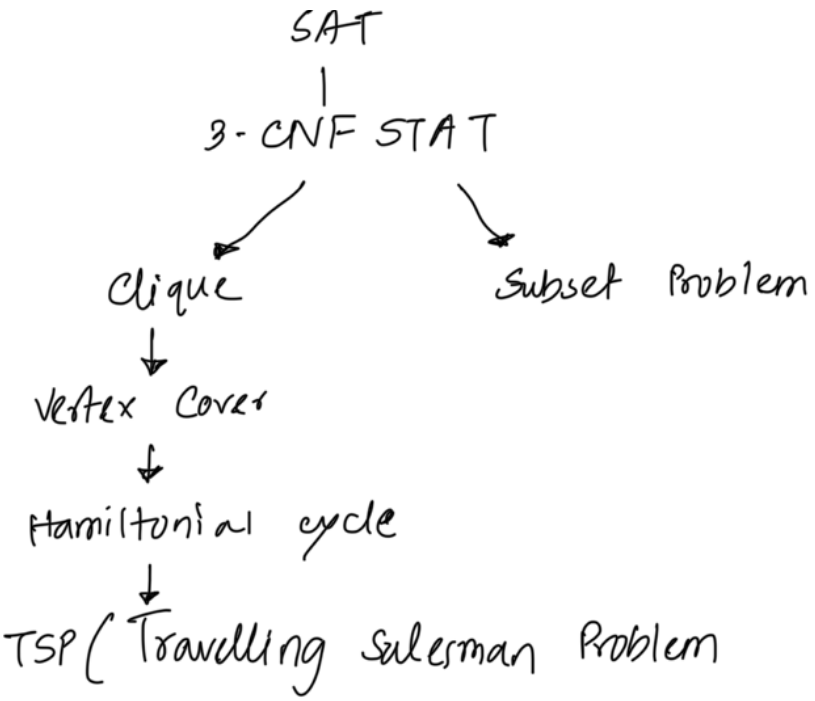
PSPACE

- polynomial space

6 November, 2024

Circuit - Satisfiability

!



3-CNF-SAT $\leq_p$ Clique

- একটা graph থেকে clique subset বের করার কোনো naive solution নাই in polynomial time. Exponential time এ হয়।
- clique problem belongs to NP.

(How? কেউ একজন আমাকে একটা non-deterministic solution দিয়েছে, we can verify that in polynomial time)

[verification যদি polynomial time এ হয় তবে NP]

Poove that clique problem is NP-hard.

How? Already established একটা NP problem কে এখানে reduce করব।

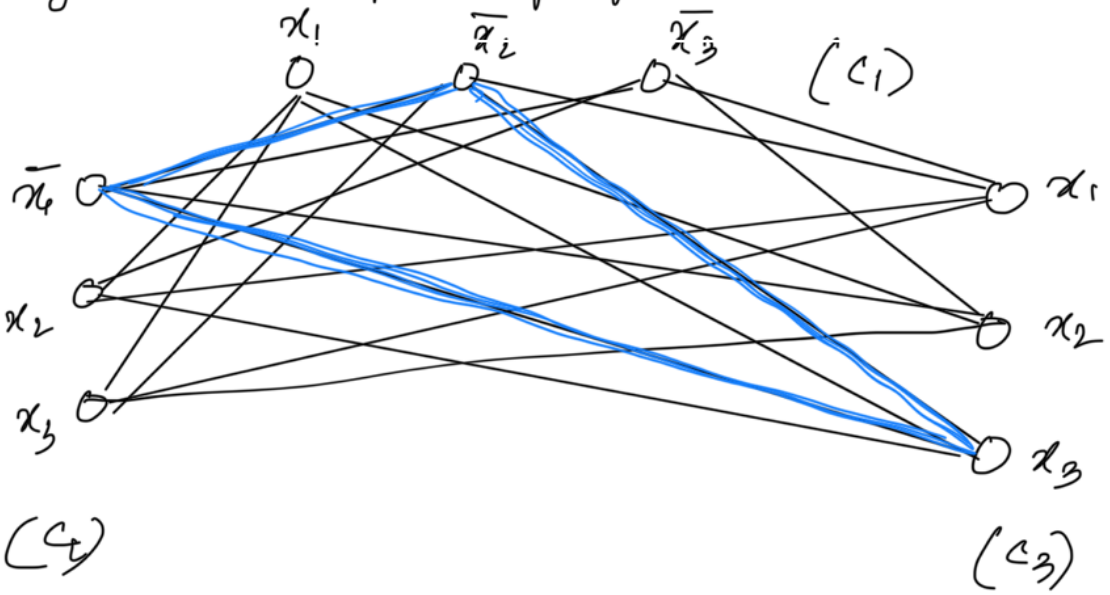
$$\varphi_1 = c_1 \wedge c_2 \wedge c_3$$

$$c_1 = (x_1 \vee \overline{x_2} \vee x_3)$$

$c_2 = (\bar{x_1} \vee x_2 \vee x_3)$

$c_3 = (x_1 \vee x_2 \vee x_3)$

Drawing the corresponding graph:



(x_1 এর সাথে x_2, x_3 দুটো connect হবে and so on.

Identify a clique of size 3: $\bar{x_1}$ $\bar{x_2}$ x_3

If we assign : 0 0 1

(0 0 = bar, 1 = not bar)

This value assignment will satisfy the expression. Since reducible, so clique is NP complete. (?) আবার NP hard

VC is an NP complete problem → Prove.

(We have to reduce an NP hard problem to VC)

First we have to say,

$VC \in NP$

Then we have to say:

VC ∈ NP-hard

Then we have to reduce an NP hard problem to VC. Then it will be NP complete.

Hamiltonian cycle এর টা নিজে পড়তে হবে.
TSP will have to be proved by Ham cycle.

We have proven a problem is NP complete.
Now what?
→ We will try to give an approximation algorithm+ approximation ratio
↓
আমার result optimized result এর চেয়ে কত গুণ খারাপ.

For minimization problems, maximization:

$$\frac{\text{our output}}{\text{optimal output}} = \frac{C}{C^*} \qquad \frac{\text{Optimal output}}{\text{Our output}} = \frac{C^*}{C}$$

Matching Problem

Set of edges যাদের মধ্যে common vertex নাই.
Maximization problem: we want the maximum number of edges.

Maximal বলার সময় context বলে দিতে হবে.

Context expand করলে আর maximal না-ও হতে পারে, So, maximal is not always maximum.

- Vertex Cover বের করার একটা polynomial time এর algorithm খুঁজে বের করব,
  Let যেই vertex এর সাথে বেশি edges থাকে আগে pick করব। Pick করতে থাকবো until we get all the edges.
  Now, find the ratio mathematically.

Let, যতগুলো vertex pick করলাম, Then the ratio will become $\frac{n}{c^*}$ (very bad)

- Ratio 2, ½, 5 এরকম হলে OK.

Matching এর algorithm দিয়েও Vertex Cover বের করা যায় (বই এ আছে)
→ এটার approximation ratio 2.

Let, we have a set of edges of cardinality A (কতগুলো edge আছে সেই set এ). For those edges, the optimal vertex cover be C^*

$$C^* \geq A$$

and $C = 2|A|$ (আমার approach এর vertex cover)

$$\therefore C^* \geq \frac{C}{2}$$

$\therefore C \leq 2C^*$

TSP (Travelling Salesman Problem)

একটা শহর থেকে travel শুরু করবে। He has to visit all the cities and return to original city in minimum cost.

Think of an approximation approach for this.

→ First find the minimum spanning tree (e.g. Kruskal's algorithm) → we won't lose any vertices.

→ Then add one edge to make it a cycle.

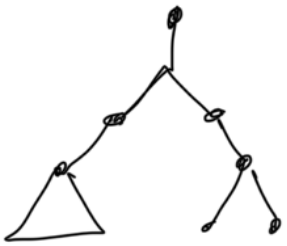
$\therefore$ Optimal size = size of minimum spanning tree + 1.

সবগুলো vertex visit করলে একটা walk হবে।

এখানে we put the restriction of triangle law:

The sum of any two edges > third edge.

[Triangular Inequality]



সবগুলো ঘুরার জন্য we have to return from some as well

$\therefore W = 2(H)$

cost for cycle — cost of walk

11 November, 2024

Assignment 2 discussion:

- subgraph isomorphism problem এর solution polynomial time এ verifiable by using adjacency list/matrix.
- verification এ size দেওয়া থাকলে সবার আগে size check করব, size match না করলে বাকি condition আর check করার দরকার নাই.

- Approximation of Travelling Salesman problem
- Approximation of Vertex Cover problem

) Common

Data Structures

Hash / hash table

- insert, delete, update, সব করতে চাই constant time এ. access/search [O(1)]

This is possible in array. Then why do we need hash?

We want to work with a dynamic set [আমাকে ৫টা জিনিস আছে, কালকে ৭টা হবে...]

- hash এর key is a small subset of all possible values. We have to map those key values to all possible values.

Hash function

Input: যেই value কে রাখতে চাই

Output: we will get the index যেখানে রাখবো.

common hash function: mod

এখানে একই index এ অনেকগুলো element হতে পারে → collision. There are various methods to resolve collision.

1. By chaining (with pointers)

→ table এ কোনো value রাখবো, bucket এর pointer থাকবে।

এই মেথডে constant time এ insert করতে পারবো but search করতে পারবোনা।

We want to design a hash function which will evenly distribute all values (does not happen perfectly irl)

2. Open addressing: (Table এই data রাখবে, not pointer)

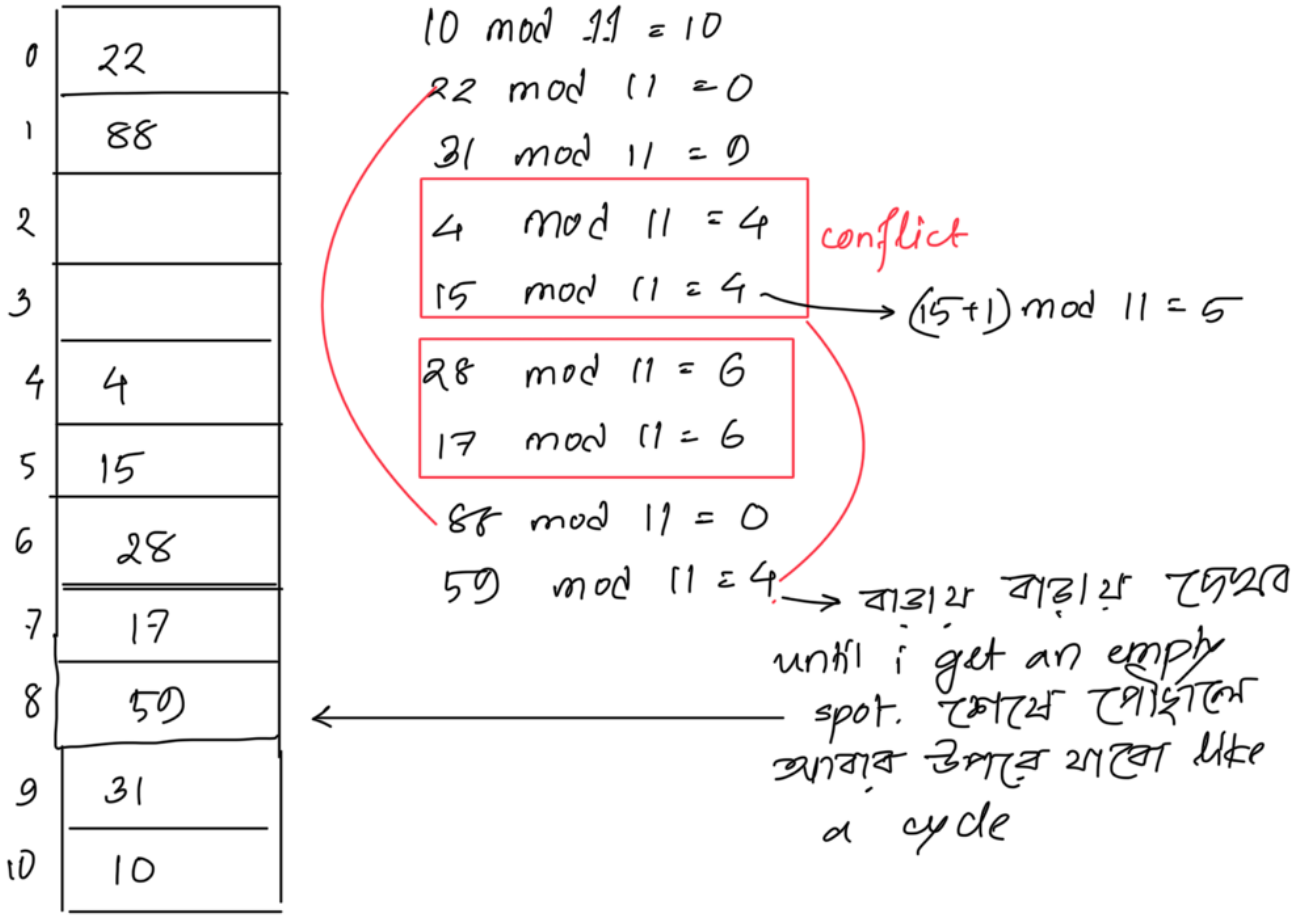
Table will be as big as required to accomodate data.

Table full হয়ে গেলে rehashing করবো।

এখানেও collision হতে পারে। To resolve it: Probing

1. Linear probing
   → results in clustering problem
2. Quadratic probing
   → jump দিয়ে দূরে যাবে
3. Double hashing: দুইটা hash function থাকবে

Q. Exercise 11.4-1 (i will be 0, 1, ... gradually)



(double hashing part বুঝিনাই, ask in next class and read book)