Solutions for End Sem. Paper

CS204: ALGORITHMS

Part I

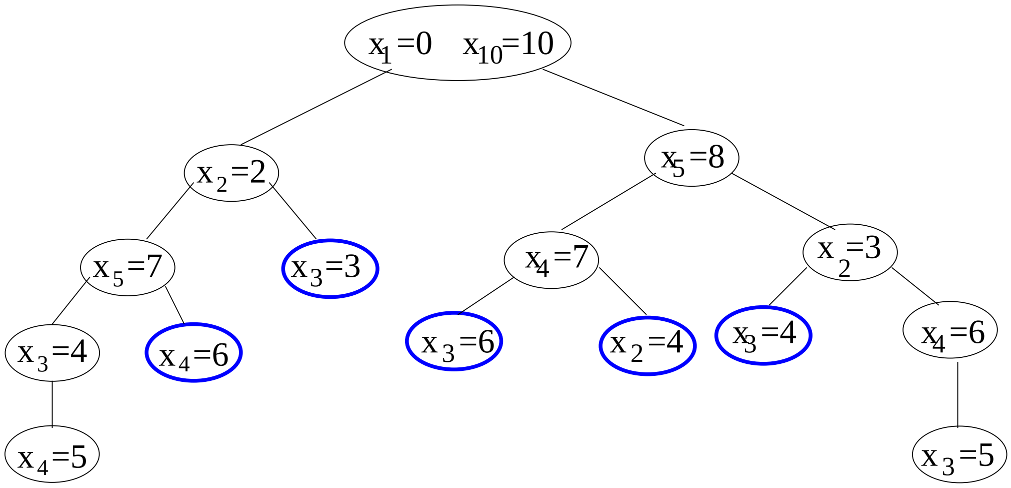
Question No.	1	2	3	4	5	6	7	8	9	10
Answer	C	B	D	B	A or C	A	A	B	A	C
Question No.	11	12	13	14	15	16	17	18	19	20
Answer	D	B	F	A	C	D	E	B	A	F

Part II

1. Construct the search tree that the backtracking technique will generate for the instance of the turnpike reconstruction problem specified by the set of distances [7]

$$D = \{1, 2, 2, 2, 3, 3, 3, 4, 5, 5, 5, 6, 7, 8, 10\}$$

Answer: The tree will be as follows:



2. Recall that a non-deterministic computational model is an augmented RAM model with three additional instructions.

   i. `choice(S)`: arbitrarily choose one element from input set S

   ii. `failure()`: signals unsuccessful termination of the current thread.

   iii. `success()`: signals successful completion of the current thread and forces termination of all other thread.

Write $\mathcal{NP}$ algorithm for finding a clique of size k of an input graph G. Determine the running time. [9]

Answer: The algorithm is as follows:

```
cliqueAlgorithm(G, k) {
    S = Φ;
    for (i = 1; i ≤ k; i++) {
        v = choice(1:n);
        if (v ∈ S) then failure();
        S = S ∪ {v}
    }
    for ((each pair (u, v) ∈ S) && (u ≠ v))
        if (u, v) ∉ E then failure();
    success();
}
```

3. Consider the divide and conquer algorithm for closest pair problem discussed in the class. It first splits the space into two halves by a vertical line and recursively solves closest pair in each half space. Let d be the minimum of two values obtained. Then it considers a strip of width $2d$ around the splitting line as indicated by the figure 1. Prove that corresponding to each point on the left side of the split line there can be at most 6 points on the right side which may be at a distance less then d. [8]

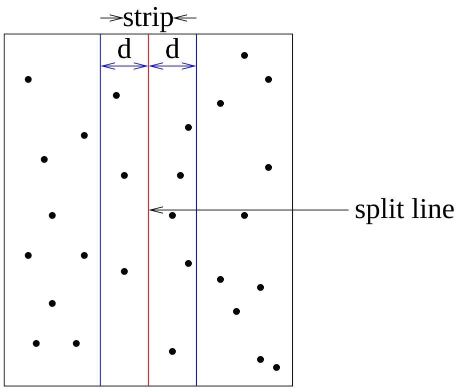
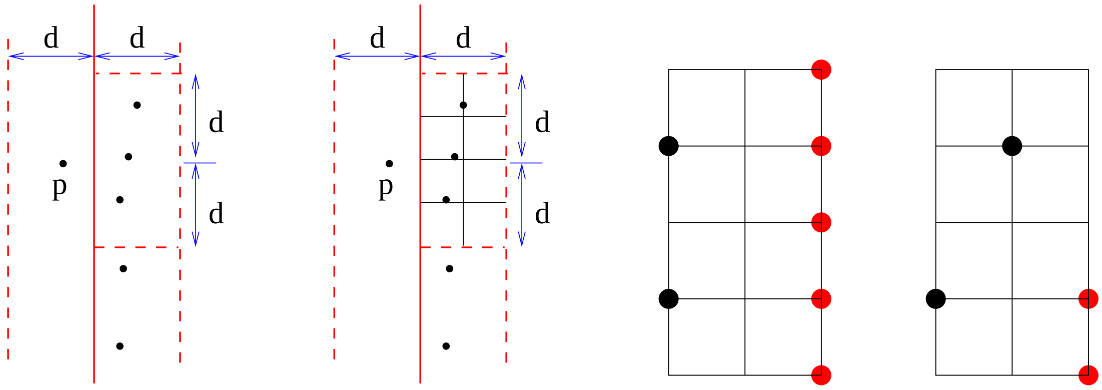


Figure 1: Strip along split line (question 3)

Answer:



Consider a point p on the left half of the strip of the leftmost picture in the figure shown above. Corresponding to p we prove that only 6 points are required to examined on the right half of the strip. The points to be examined must lie within the rectangle as indicated.

Let us divide the rectangle into 8 squares of size $(d/2) \times (d/2)$ as in the middle picture in the figure. The length of the diagonal of each such square is $d/\sqrt{2} < d$. Hence, no more than one point can exist in each of 8 squares. But the points of two adjacent middle squares either to the left or to the right must coincide. Otherwise, the points in each such square must be placed in far corners, which will violate distance restriction with respect to points in other adjacent squares as shown by the rightmost picture in the figure above.

Hence a maximum of 6 points need to be considered for each point p on the left. We can presort the points in left half strip and do the walking over the sorted list. Since there can be at most n points in a half strip, the searching time for closest along the middle strip can require at most $O(n)$ time.

4. A k-tape Turing Machine shown in figure 2. It consists of k tapes each of which extends to infinite length to right. It has k tape heads, each of which moves independent of the other either one position to left (L) or to right (R) or stays stationary (S). A new state of the finite control is determined by each move. Now consider the next move function for a 2-tape TM tabulated below and determine what language it accepts? [16]

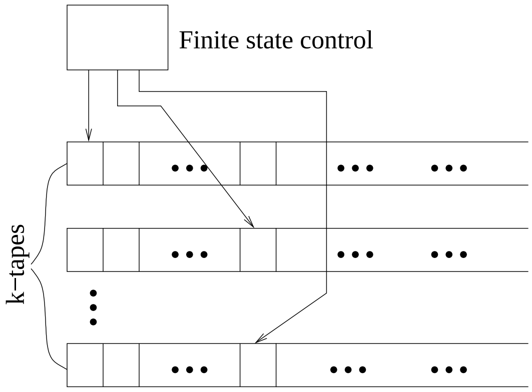


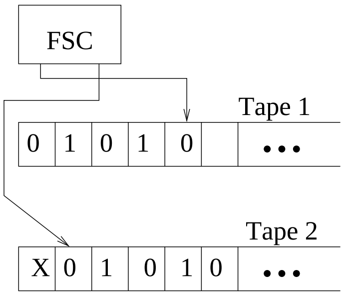
Figure 2: A k-tape TM (question 1)

Current	Symbol on:		New symbol,move		New	
state	Tp 1	Tp 2	Tp 1	Tp 2	state	Comments
q_0	0	b	0,S	X,R	q_1	If input is nonempty, print X on
	1	b	1,S	X,R	q_1	tape 2 and move head right; go to
	b	b	b,S	b,S	q_5	state q_1. Otherwise, go to q_5.
q_1	0	b	0,R	0,R	q_1	Stay in state q_1, copying tape 1
	1	b	1,R	1,R	q_1	onto tape 2 until b is reached
	b	b	b,S	b,L	q_2	on tape. Then go to state q_2.
q_2	b	0	b,S	0,L	q_2	Keep tape 1's head fixed and
	b	1	b,S	1,L	q_2	move to tape 2's left until X is
	b	X	b,L	X,R	q_3	reached. Then go to q_3.
q_3	0	0	0,S	0,R	q_4	Compare symbols on two tapes
	1	1	1,S	1,R	q_4	move tape 2's head right and
						go to q_4.
q_4	0	0	0,L	0,S	q_3	In q_4 go to q_5 and accept
	0	1	0,L	1,S	q_3	if head has reached b on tape 2.
	1	0	1,L	0,S	q_3	Otherwise move tape 1's head left
	1	1	1,L	1,S	q_3	and go back to q_3. Alternation q_3,
	0	b	0,S	b,S	q_5	q_4 prevents input head from going
	1	b	1,S	b,S	q_5	beyond leftmost cell of tape 1.
q_5						Accept

Answer: The TM program specified by next move function tabulated above is for recognition of palindrome. All bit strings which are palindromes are accepted by this function. Consider, for example, the string 01010. The state transitions will be as follows:

- The symbol 0 in tape 1 is scanned then X is written on first cell of tape 2 and the TM enters q_1 state.

- TM remains in q_1 state, each time a new symbol scanned tape 1's head move one position to right and the symbol read is written back to the same cell in tape 1 and also written on the tape 2 with the latter's head moving to right. This is continued until the entire string 01010 has been scanned and processed similarly. After this step tape 2 contains the same string preceded by X in the leftmost cell.
- When a blank symbol is encountered on tape 1, tape 2 head writes a blank and moves to one position left and the TM now enters q_2 state.
- TM stays in q_2 and every time tape 2's head moved 1 position to left until encountering X on it. During this, tape 1's head does not move at all. When X is encountered on tape 2, tape head 1 moves to left and tape head 2 moves to right writing back their respective symbols and the TM enters q_3 state. So at the beginning of state q_3 the situation is as shown below.



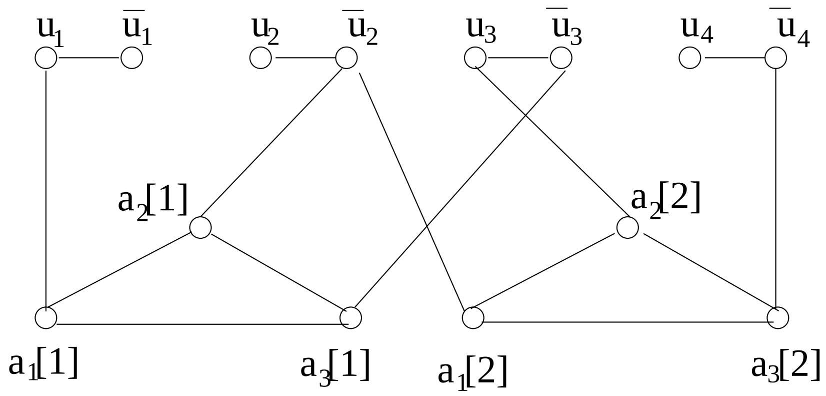
- TM enters q_4 after comparing two symbols in two tapes.
- In q_4 head 1 advance to left and again the TM enter q_3 state.
- So now TM loops between q_3 and q_4 until in a q_4 state head 2 encounters blank then TM enters accepting state q_5.
- It means this TM will only accepts palindromes but strings which are not palindromes, its moves are not defined.

5. Construct the graph for the following instance of 3-SAT problem

$$\begin{aligned} U &= \{u_1, u_2, u_3, u_4\} \\ C &= \{\{u_1, \bar{u}_3, \bar{u}_2\}, \{\bar{u}_2, u_3, \bar{u}_4\}\} \end{aligned}$$

so that if the 3-SAT instance is satisfiable then the graph has a vertex cover size 8.[10]

Answer:



6. Prove that for a given graph $G = (V, E)$ and a subset $V' \subseteq V$ following are equivalent.

   (a) V' is a vertex cover for G.

   (b) $V - V'$ is an indepenedent set for G.

   (c) $V - V'$ is a clique for the complementary graph $G^c = (V, E^c)$, where

$$E^c = \{(u, v) \mid u, v \in V \text{ and } (u, v) \notin E\}.$$

[10]

Answer: Let V' be a vertex cover for $G = (V, E)$. Now consider $V - V'$. Let u, v are any two vertices $V - V'$. If (u, v) is an edge in G then either u or v must belong to V'. This implies no pair of vertices in $V - V'$ are adjacent. So $V - V'$ is an independent set in G.

Consider, the complementary graph G^c. There is an edge between u and v in G^c if and only if $(u, v) \notin E$. So, every pair of vertices of an independent set of G must be connected in G^c. That is $V - V'$ is a clique in G^c.

Now to end the proof, consider a set of vertices V' which defines a clique in G^c. Let (u, v) be an edge in E which means both u and v cannot belong to V'. So, there are two possible cases:

case 1: u or v belongs to V'.

case 2: neither u nor v belongs to V'.

In both cases, one of u or v must belong to $V - V'$, so the edge (u, v) is covered.