1) Bubble sort and its variant.

a) There are exactly 2 variant of bubble sort.

b) The sequence to be sorted. Given a sequence $\langle a_1, a_2 \ldots a_n \rangle$ the algorithm first performs $n-1$ compare exchange operation.

c) An iteration of the inner loop of bubble sort takes time $O(n)$ and we perform $O(n)$ iteration so total complexity $O(n^2)$

d) Bubble sort is difficult to parallize algorithm :-

```
BUBBLE_SORT (n)
    begin
        for j = n-1 to 1 do
            for i = 1 to j do
                compare_exchange (aj, aj+1);
    end BUBBLESORT
```

Odd even transposition.

a) The odd even transposition sorts n elements in a phase each phase require n/2 time.

b) There are 2 phases 1) odd
2) even

c) In Odd phase, element with odd indices are compared to adjacent elements.

d) Some of even phase but with just even indices.

e) After n phases of odd even phase sequence is sorted.

f) Complexity is $O(n^2)$

2) Parallel Depth-first Search

1) ~~Parallel~~ Depth first search algorithms solve DOP's that can be formulated as tree search problems.
2) DFS begins by expanding the initial node & then generating its successors.
3) In each subsequent step, DFS expands one of the most recently generated nodes.
4) If this node has no successors (or cannot lead to any solutions) then DFS backtracks and expands a different node.
5) A major advantage of DFS is that its storage requirement is linear in the depth of the state space being searched.

The following sections show three algorithms based on depth first search.

3) Parallel best first Search

1) In most parallel formulations of BFS, different successors concurrently expand different nodes from open list.
2) These formulations differ according to the data structures the use to implement the open list. Given p processes, the simplest strategy assigns each processors to work on one of the current best nodes on the open list
3) The parallel BFS expands more than one node at a time. it may expand nodes that would not be expanded by a sequential algorithm.
4) The termination for criteria for normal BFS fails for parallel BFS.
5) Since the open list is the accessed for each node expansion, it must be easily accessible to all processors which will decrease efficiently.

c) Following are steps of parallel Best first search.

a) When a node is generated, it is sent to the processor, whose label is returned by the hash function for the node.

b) Upon receiving the node, a processor checks, whether it already exists in the local open or closed list.

c) If not, the node is inserted in the open list.

d) If the node already exists, and if the new node has a better cost associated with it, then the previous version of the node on the open list.