Theory Assignment-5: ADA Winter-2024

Ritika Thakur (2022408) Sidhartha Garg (2022499)

1 Preprocessing

1. **Graph Representation**: Represent each box as a pair of nodes A and $A+n$ where n is the total nodes.
2. **Determine Nesting Relationship**: Define a function to check if box $(A+n)th$ can be nested in box $(A)th$. Add a directed edge from A to $(A+n)th$ with capacity 1 if box $(A+n)th$ can be nested inside box $(A)th$. A box is said to be nested inside another box if the dimensions of that 'nested' box are strictly less than the dimensions of the parent box i.e. if box A can be nested in box B then each side of box A should be less than at least one unique side of box B.
3. **Capacity Assignments**:
   - Add a directed edge from the super source to each A with capacity 1.
   - Add a directed edge from each $A+n$ to the super sink with capacity 1.
4. **Building the Graph**: Construct the graph based on the above rules. Use the graph for maximum flow computation and find the maximum matching for the bipartite graph.

   Time Complexity for building the graph:

   - **Graph Representation**: Creating $2n$ nodes is $O(n)$.
   - **Determine Nesting Relationship**: Checking pairs of boxes for nesting requires $O(n^2)$.
   - **Capacity Assignments**: Adding edges to/from source and sink requires $O(n)$.
   - **Overall Time Complexity**: Dominated by $O(n^2)$.

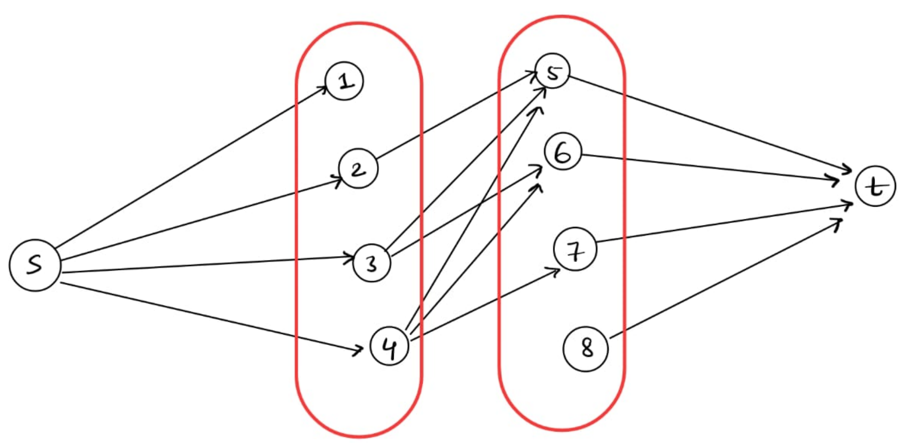

Figure 1: A simple preprocessed graph for 4 boxes such that 1 fits in 2 fits in 3 fits in 4

2 Algorithm Description

The algorithm consists of the following steps:

1. **Run Maximum Flow Algorithm:**
   - Run any max flow algorithm from the super source to the super sink.
   - The Ford-Fulkerson algorithm is one possible choice.
2. **Interpret the Maximum Flow:**
   - The maximum flow value represents the maximum number of times you can nest one box inside another.
   - This helps reduce the number of visible boxes.
3. **Calculate the Minimum Number of Visible Boxes:**
   - Calculate the minimum number of visible boxes as:

$$\text{number of boxes(n)} - \text{max flow}$$

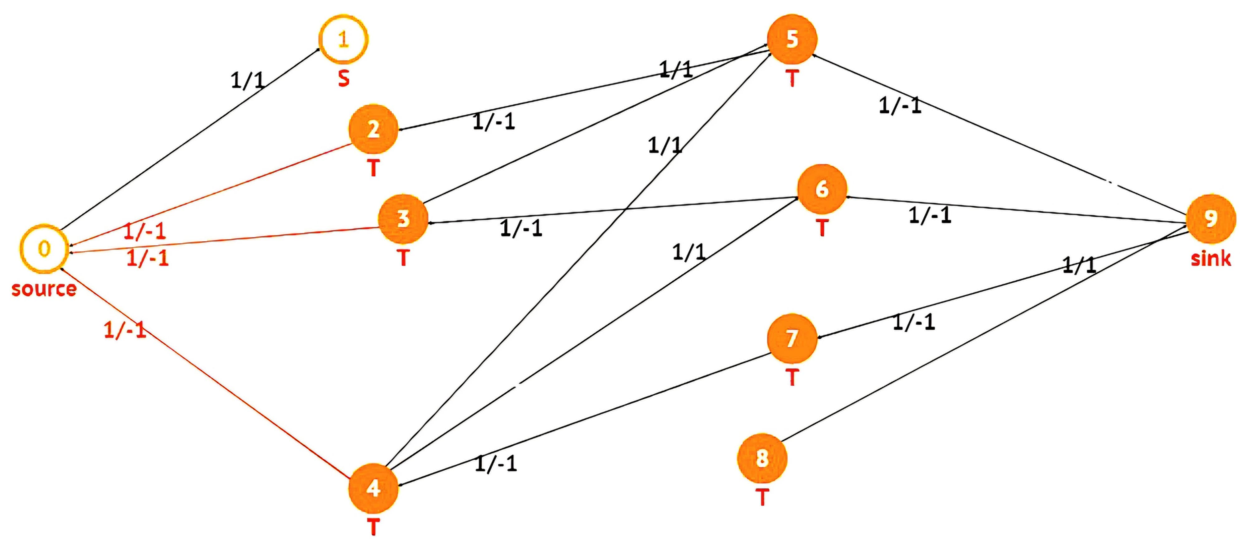

Figure 2: Running the algo on the above graph. ref: visualgo.net used for creating visualisation

3 Proof of Correctness

To prove the correctness of the algorithm, we need to show that:

1. The algorithm always terminates.
2. The maximum flow obtained corresponds to the maximum number of nested boxes.
3. The minimum number of visible boxes is correctly calculated.

Termination:

The algorithm consists of constructing a graph and then running a maximum flow algorithm, which is guaranteed to terminate. Thus, the algorithm always terminates.

Maximum Flow Corresponds to Nested Boxes:

- By construction, we create a directed graph where each box is represented by two nodes, one for its start and one for its end.
- We add edges between boxes such that a box can be nested inside another if the corresponding edge exists in the graph.
- Running a maximum flow algorithm finds the maximum matching in this bipartite graph.
- Each edge in the maximum matching corresponds to nesting one box inside another.
- Therefore, the maximum flow obtained corresponds to the maximum number of nested boxes.

Minimum Number of Visible Boxes Calculation:

- The minimum number of visible boxes is calculated as the total number of boxes minus the maximum flow.
- Since the maximum flow represents the maximum number of nested boxes, subtracting it from the total number of boxes gives the minimum number of visible boxes.
- Therefore, the calculation of the minimum number of visible boxes is correct.

Conclusion:

The algorithm terminates, and it correctly computes the maximum flow, which corresponds to the maximum number of nested boxes. It also accurately calculates the minimum number of visible boxes. Therefore, the algorithm is correct.

4 Complexity Analysis

4.1 Time Complexity

In a complete graph with n nodes, there are $\frac{n(n-1)}{2}$ edges. Using the Ford-Fulkerson algorithm, you can add $2n$ additional edges to connect the source and sink. So, the total number of edges in the graph is approximately n^2.

The maximum flow in the network can be around n, as it represents the upper limit for the size of any matching.

Therefore, the product of the number of edges and the maximum flow is:

$$n^2 \times n = n^3$$

This is the time complexity of the Ford-Fulkerson algorithm in this scenario.

4.2 Space Complexity

The space complexity of the algorithm involves storing the graph with $2n$ nodes plus the source and sink, and up to n^2 edges between the nodes. This requires $O(n^2)$ space. Additionally, the max flow algorithm uses space for tracking paths and flows, also around $O(n^2)$. Thus, the overall space complexity is $O(n^2)$.

5 Pseudocode

Algorithm 1 Max Flow for Box Nesting

```
procedure BuildGraph(Boxes)
    Create a directed graph G
    for i from 1 to n do
        Add nodes A_i and A_{i+n} to G
    end for
    for i from 1 to n do
        for j from 1 to n do
            if i ≠ j and Box (A_{i+n}) can be nested in (A_j) then
                Add edge from A_i to A_{j+n} with capacity 1
            end if
        end for
    end for
    for i from 1 to n do
        Add edge from source to A_i with capacity 1
        Add edge from A_{i+n} to sink with capacity 1
    end for
end procedure
procedure RunMaxFlow
    Run Ford-Fulkerson Algorithm from source to sink
end procedure
procedure InterpretMaxFlow
    The maximum flow represents the maximum number of times one box can be nested inside another
end procedure
procedure CalculateMinVisibleBoxes(number of boxes (n), max flow)
    Minimum number of visible boxes ← n − max flow
end procedure
```

6 References

- Math Stack Exchange