**WORKSHEET 17**

**Lecture on Counting Sort, Topological Sort, Quickselect**

|  |
| --- |
| PART 1. Counting & Topological Sort |

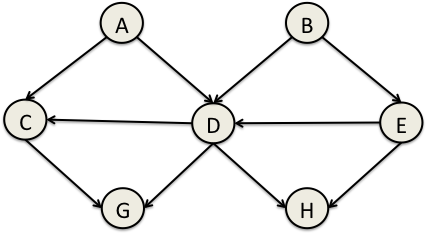
1.1. This array stores the following information:

* + Position (index) k registers how many times number k appears in a set.
  + For example, number 0 appears once, number 8 is not in the set

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2 | 0 | 0 | 1 | 2 | 1 | 0 | 1 |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |

Please, reconstruct the original set of numbers sorted from smallest to largest.

1.2 Find a feasible topological sort for this graph:



One possible sort:

|  |
| --- |
|  |

1.3 For the following graph, execute the Depth-First Topological Sort:

|  |  |
| --- | --- |
|  | Final topological sort: |

Remember to draw the contents of list L and sets U,T and P as you execute the algorithm.

|  |
| --- |
| PART 2. Selection |

2.1 Please, complete the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data structure for maxes** | **smalles(maxes)**  **(time complexity)** | **remove\_min(maxes)**  **(time complexity)** | **insert(A[i],maxes)**  **(time complexity)** |
| **Unsorted array** |  |  |  |
| **Sorted array** |  |  |  |
| **Min-heap** |  |  |  |