Data Structures and Algorithms Spring 2024 — Problem Sets

by Kirill Efimovich

June 24, 2025

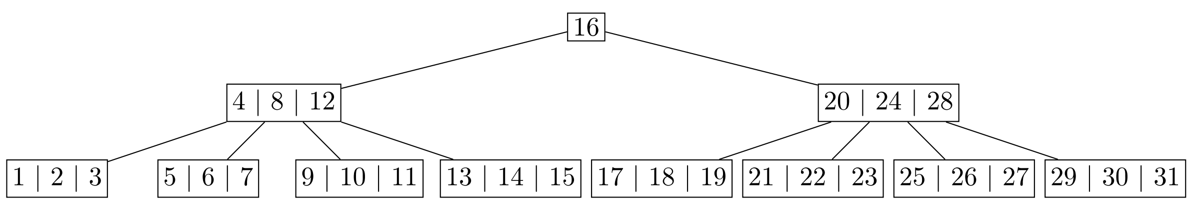
Week 15. Problem set

1 Task 1

1.1 Statement 1

(+1.5 % extra credit) Answer the following questions about B-trees:

(a) Following the definition of B-trees [Cormen, §18], which values of the minimum degree t are valid for this B-tree? Provide a brief justification.



(b) What is the maximum possible number of keys stored in a B-tree with minimum degree 5 and height 9. Height here is the maximum number of edges in a path from the root to some leaf (so singleton tree has height 0).

(c) Generalize your answer to the previous question by providing a closed form formula (and without the use of $\sum$-notation or ellipses (...)) for the maximum number of keys, stored in a B-tree with minimum degree t and height h. Provide full justification for your formula.

1.2 Solution

Answer:

(a): The root node can contains 1 to $2t-1$ keys, and non-root nodes contains $t-1$ to $2t-1$ keys. In this B-tree, each node at levels 2 and 3 has 3 keys. Hence, each non-root node has at least $t-1 = 2-1 = 1$ key and no more than

$2t-1 = 2\cdot 2-1 = 3$ keys. Hence, the minimum absolute degree t of this B-tree is 2.

(b): The maximum number of keys that can be stored in a B-tree with a minimum degree t and height h can be found by the formula $2t^{(h+1)} - 1$. For our $h = 9$ and $t = 5$, we get $(2\cdot 5)^{(9+1)} - 1 = 10^{10} - 1 = 9999999999$. Hence, maximum number of keys that can be stored in a B-tree with a minimum degree of 5 and a height of 9 is 9999999999.

(c): The formula for the maximum number of keys in a B-tree with height h and minimum degree t is $2t^{h+1} - 1$. The root node in a B-Tree can contain between 1 and $2t-1$ keys. Every level of the tree adds twice as many keys as the previous level, because each key in a node has a child associated with it, except for the leaves. Therefore, the total number of keys is calculated by adding up the number of keys at each level, starting with the root and going down to the leaves, and then subtracting 1 for the final node. Final formula:

$\sum_{i=0}^{h} 2t^i \cdot (2t-1) = (2t-1)\cdot \sum_{i=0}^{h}(2t)^i = (2t-1)\frac{(2t)^{h+1}-1}{2t-1} = (2t)^{h+1} - 1$

2 Task 2

2.1 Statement

(+1.5 % extra credit) Give an example of a connected weighted undirected graph $G = (V, E)$ with a selected initial vertex, such that:

- G has exactly 5 vertices ($|V| = 5$) named A, B, C, D, E and exactly 9 edges ($|E| = 9$)
- each edge in E has a positive integer weight in the range from 1 to 9
- the minimum spanning tree built by the Prim's algorithm from the selected initial vertex is unique (does not depend on the order in which edges are considered)
- the shortest paths tree built by the Dijkstra's algorithm from the selected initial vertex in unique (does not depend on the order in which edges are considered)
- the number of successful Decrease-Key operations in Prim's algorithm is different from the number of successful Decrease-Key operations in Dijkstra's algorithm, starting from the selected initial vertex.

For the justification:

(a) Provide the **graph G** (clearly rendered).

(b) Clearly specify the **initial vertex**.

(c) Provide the **unique tree** that is both the unique minimum spanning tree produced by the Prim's algorithm and the shortest paths tree produced by the Dijkstra's algorithm.

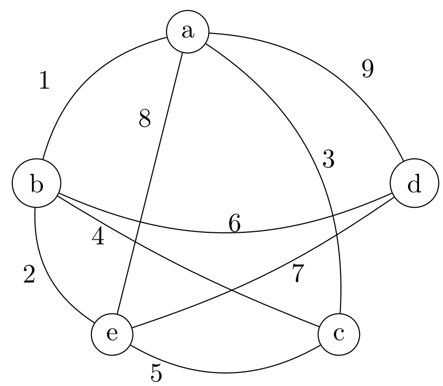
(d) Write down the **sequence of calls** and the **total number of successful calls** to Decrease-Key in the Prim's algorithm, clearly indicating which are successful.

(e) Write down the **sequence of calls** and the **total number of successful calls** to DECREASE-KEY in the Dijkstra's algorithm, clearly indicating which are successful.

2.2 Solution

Answer:

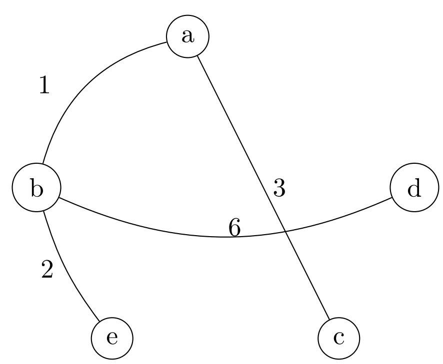
(a):



(b):

Initial vertex: **A**.

(c):



(d):

DECREASE-KEY calls in the Prim's algorithm:

1. Decrease-Key(**B**, 1) - Successful (Edge AB)
2. Decrease-Key(**E**, 2) - Successful (Edge BE)
3. Decrease-Key(**D**, 6) - Successful (Edge BD)
4. Decrease-Key(**C**, 3) - Successful (Edge AC)

Total number of calls: 4.

(e):

DECREASE-KEY calls in the Dijkstra's algorithm:

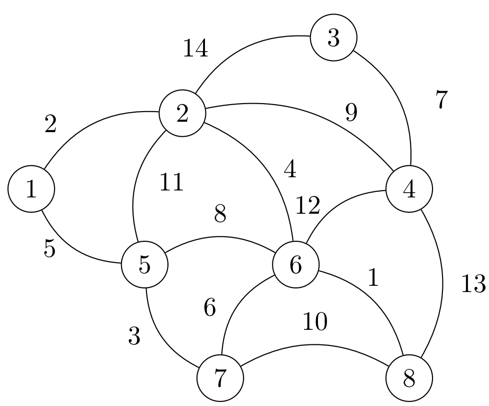
1. Decrease-Key(**B**, 1) - Successful (Edge AB)
2. Decrease-Key(**E**, 2) - Successful (Edge BE)
3. Decrease-Key(**D**, 6) - Successful (Edge BD)
4. Decrease-Key(**C**, 3) - Successful (Edge AC)
5. Decrease-Key(**C**, 2) - Successful (Edge BEC)
6. Decrease-Key(**D**, 4) - Successful (Edge BED)

Total number of calls: 6.

3 Task 3

3.1 Statement

(+1.5 % extra credit) Run Kruskal's algorithm [Cormen, §21.2] on the following graph.

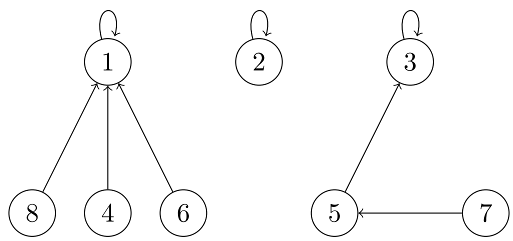


Assuming that the algorithm is using Disjoint Sets Forests [Cormen, §19.3] with union by rank and path compression heuristics, write down the sequence of operations (FIND-SET(X) or UNION(X, Y)) that are applied to the structure after its initialization. For each operation that modifies the state of the structure, demonstrate the state *after* the operation.

In the union by rank heuristic, when two representatives have the same rank, always attach the representative with the smallest index as a child to the representative with the largest index.

Each Disjoint Sets Forest state must be represented as an array, where at index i we have the index of the parent of vertex i.

For example, the following forest



Should be represented by the following array:

1	2	3	1	3	1	5	1

3.2 Solution

your solution here!

References

[Cormen] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. Introduction to Algorithms, Fourth Edition. The MIT Press 2022