RME 3111 - Lab 2: Heap

Course Instructor: Dr. Sejuti Rahman

Md. Arban Hossain (SH-092-005)

10 Feb 2023

The A* Algorithm

The A* algorithm is an informed search algorithm that uses a heuristic function to guide the search. The heuristic function is used to estimate the cost of the path from the current vertex to the destination. It is guaranteed to find the shortest path if the heuristic function is admissible. An admissible heuristic function is one that never overestimates the cost of the path from the current vertex to the destination.

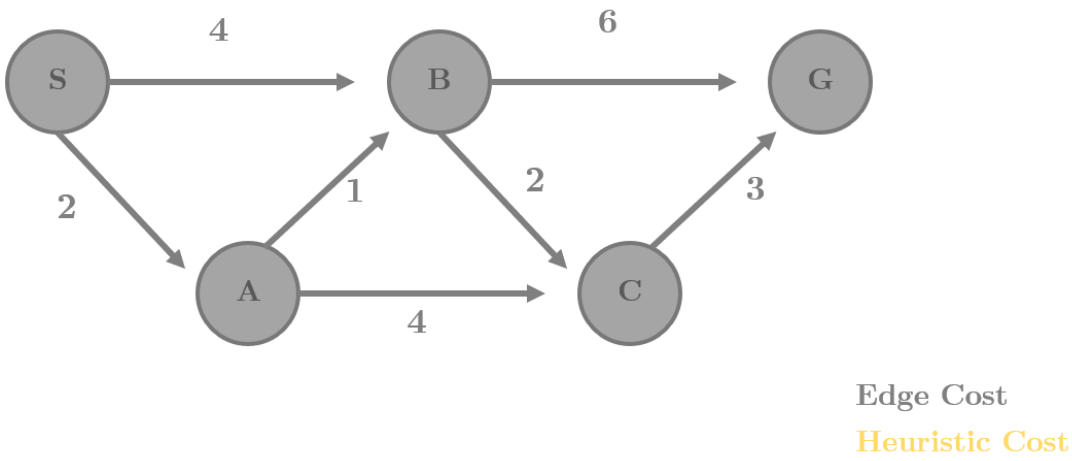
$$h(n) \leq d(n)$$

Here,
$h(n)$ = The heuristic cost of the path from the vertex to destination
$d(n)$ = The actual cost of the path from the vertex to destination
If the original cost to reach a node from the source is denoted by $g(n)$, the modified cost for the node will be,

$$f(n) = g(n) + h(n)$$

We can then run uniform cost search on the modified graph. At each step, we dequeue a node from the **Priority Queue** and check the neighbor nodes that can be reached immediately from it. If a neighbor node has a node cost that is greater than the cost of the current node plus the cost of the edge between them, we update the cost of the neighbor node and push it into the Priority Queue. We continue this process until have dequeued the destination node.

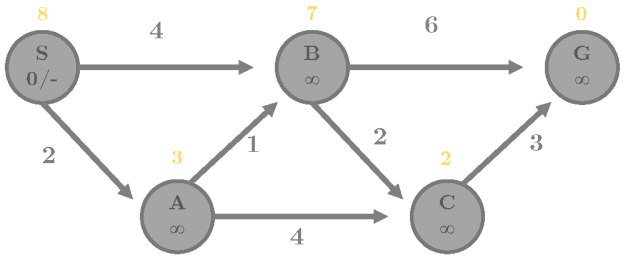
Consider the following directed graph. The vertices are labeled with their names and the edges are labeled with their costs.



Heuristic 1

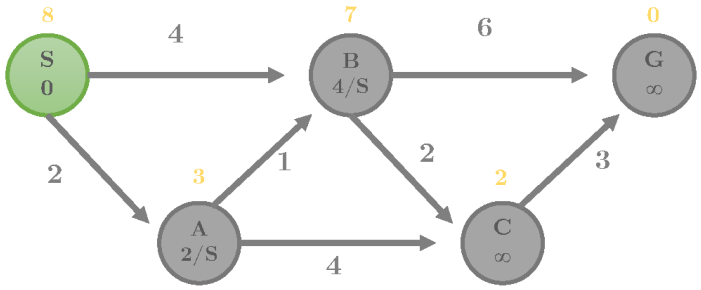
For the following set of heuristics, we can simulate the search process manually.

Vertex	Heuristic Cost
S	8
A	3
B	7
C	2
G	0



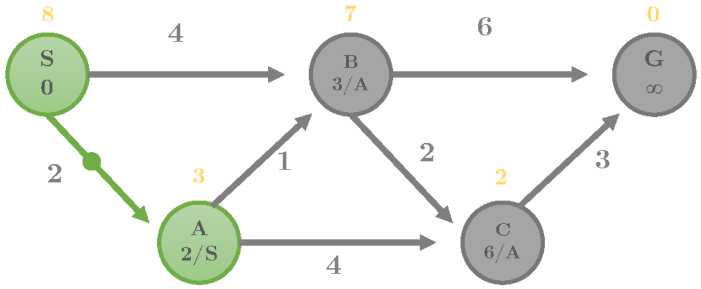
Min Heap

Node	Priority
S	0



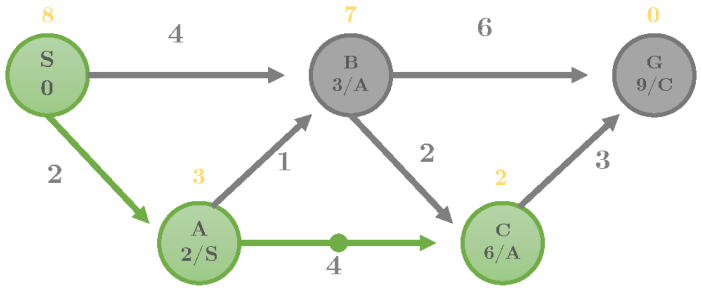
Min Heap

Node	Priority
A	5
B	11



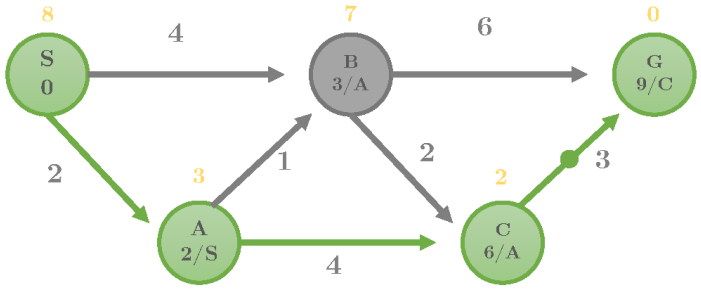
Min Heap

Node	Priority
C	8
B	10
B	11



Min Heap

Node	Priority
G	9
B	10
B	11



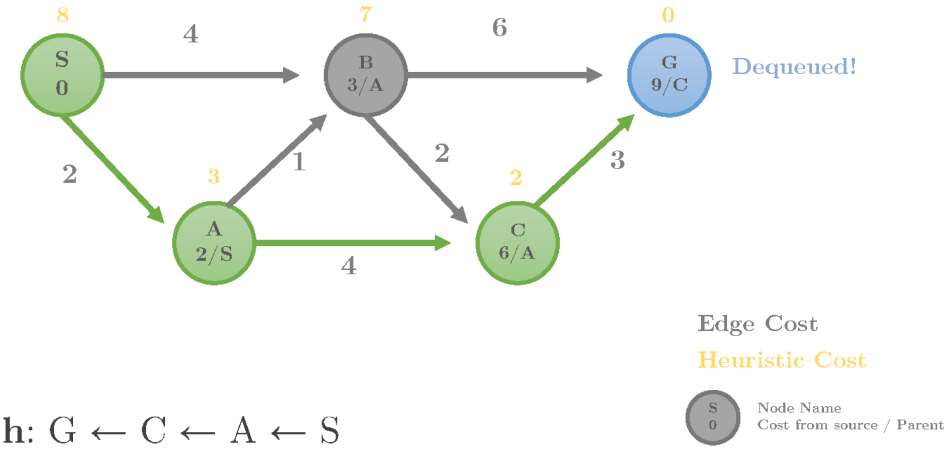
Min Heap

Node	Priority
B	10
B	11

Edge Cost

Heuristic Cost

Node Name
Cost from source / Parent



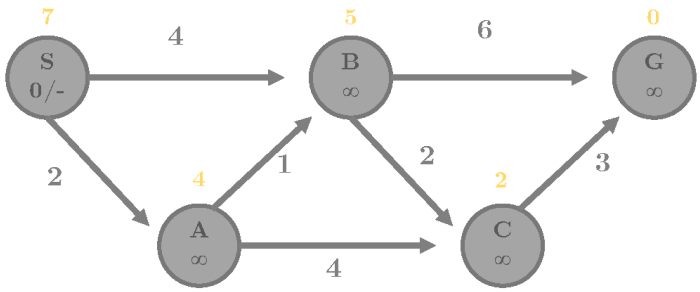
Path: G ← C ← A ← S

Path Cost: $2 + 4 + 3 = 9$

Heuristic 2

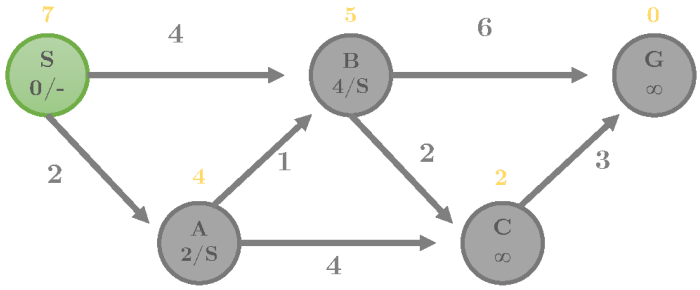
For another set of heuristics, we can simulate the search and obtain another path.

Vertex	Heuristic Cost
S	7
A	4
B	5
C	2
G	0



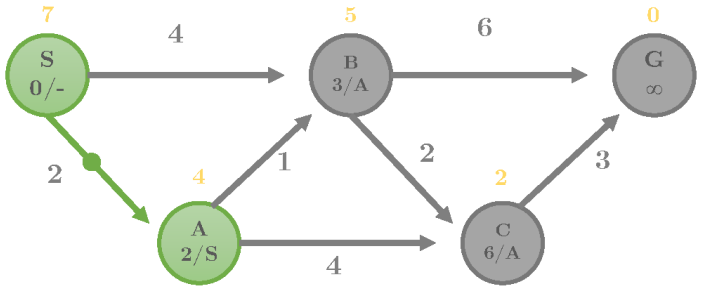
Min Heap

Node	Priority
S	0



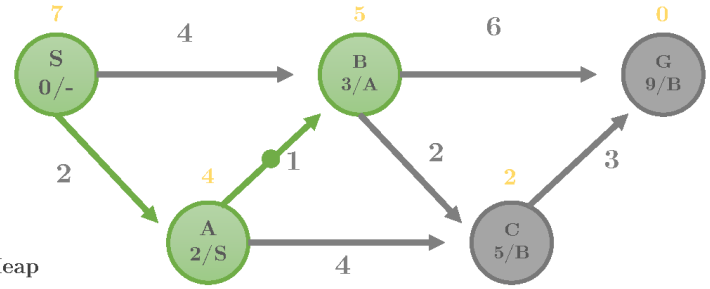
Min Heap

Node	Priority
A	6
B	9



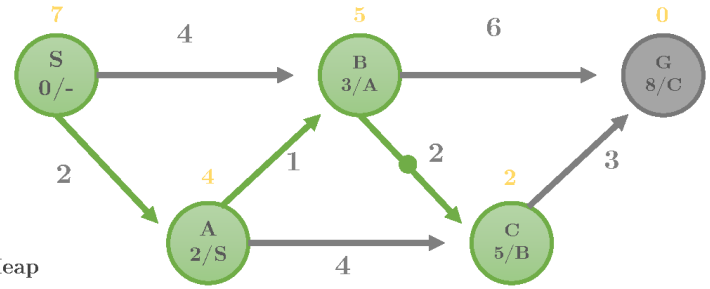
Min Heap

Node	Priority
B	8
C	8
B	9



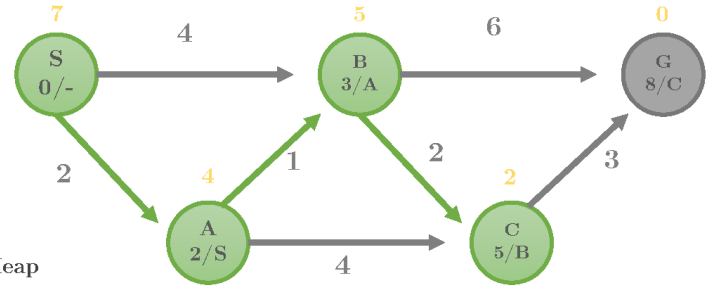
Min Heap

Node	Priority
C	7
C	8
B	9
G	9



Min Heap

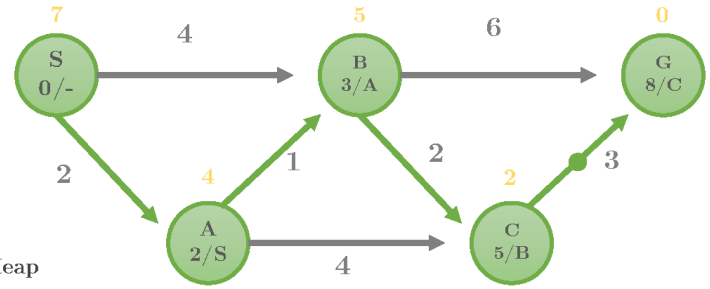
Node	Priority
C	8
G	8
B	9
G	9



Min Heap

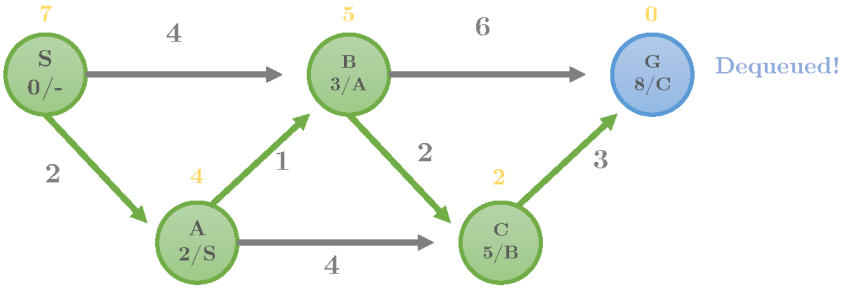
Node	Priority
G	8
B	9
G	9

Revisited C but
Nothing to push



Min Heap

Node	Priority
B	9
G	9



Path: G ← C ← B ← A ← S

Path Cost: $2 + 1 + 2 + 3 = 8$

Part B

Source Code

Attached with the report.

Output

```
(base) D:\Projects\RME_31x1\Labs\2 Heap>python Lab2.py
Using Heuristic 1:
Path Cost: 9
Path: ['S', 'A', 'C', 'G']


Using Heuristic 2:
Path Cost: 8
Path: ['S', 'A', 'B', 'C', 'G']
```

Result Discussion

The search result achieved using the second set of heuristics was more optimal (lower path cost) because the heuristic function was admissible. The first set of heuristics was not admissible because the heuristic cost of the vertex B (7) was greater than the actual cost of the path from G to B (6).