Basis. The first time the execution reaches line 3, it is the case that $i = n$. The current subarray $A[i+1, \ldots, n]$ is empty and thus, vacuously, it consists of zero in number biggest elements from $A'[1, \ldots, n]$, in sorted order. $A[1, \ldots, n]$ is a heap by Lemma 17, applied to line 1.

Maintenance. Assume the claim holds at a certain execution of line 3 and the **for** loop is to be executed at least once more. Let us call the array $A[]$ at that moment, $A''[]$. By the maintenance hypothesis, $A''[i+1, \ldots, n]$ are $n - i$ in number maximum elements of $A'[]$, in sorted order. By the maintenance hypothesis again, $A''[1]$ is a maximum element of $A''[1, \ldots, i]$. After the swap at line 4, $A''[i, \ldots, n]$ are $n - i$ in number maximum elements of $A'[]$, in sorted order. Relative to the new value of i the next time the execution is at line 3, the first sentence of the invariant holds.

The second sentence holds, too, by applying Lemma 15 or Lemma 14, whichever one is applicable (depending on whether the recursive or the iterative HEAPIFY is used), at line 6. Just keep in mind that HEAPIFY considers the heap to be $A[1, \ldots, i-1]$ because i equals A.size when the execution is at line 6; note that because of line 5, A.size is $i - 1$ at line 6. Thus at line 6, the current $A[i]$ is outside the scope of the current heap.

Termination. Consider the moment when the execution is at line 3 for the last time. Clearly, i equals 1. Plug the value 1 in place of i in the invariant to obtain "the current subarray $A[2, \ldots, n]$ consists of $n - 1$ in number biggest elements of $A'[1, \ldots, n]$.". But then $A[1]$ has to be a minimum element from $A'[1, \ldots, n]$. And that concludes the proof of the correctness of HEAP SORT. □

4.5 The Correctness of Dijkstra's Algorithm

We assume the reader is familiar with the terminology concerning weighted digraphs. When we talk about path lengths or distances, we mean weighted lengths or distances. Note the distance is not necessarily symmetric in digraphs. Let the weight function be $w : E \to \mathbb{R}^+$. The proof of correctness of the algorithm below is a detailed version of the proof in [Man05]. If $G(V, E)$ is a graph, for any $u \in V$ we denote the set $\{v \in V \mid (u, v) \in E\}$ by "adj(u)", and for any $u, v \in V$ we denote the distance from u to v in G by "dist$_G(u, v)$". The subscript G in that notation is useful when u and v are vertices in more than one graph under consideration and we want to emphasise we mean the distance in that particular graph. We postulate that dist$_G(u, v) = \infty$ iff there is no path from u to v in G.

DIJKSTRA($G(V, E)$: graph; w: weight function; s: vertex from V)

(∗ U is a variable of type vertex set ∗)
foreach $u \in V$
  dist$[u] \leftarrow \infty$
  $\pi[u] \leftarrow 0$
$U \leftarrow \{s\}$
dist$[s] \leftarrow 0$
foreach $x \in$ adj(s)
  dist$[x] \leftarrow w((s, x))$
  $\pi[x] \leftarrow s$
while $\big(\{v \in V \setminus U \mid \text{dist}[v] < \infty\} \neq \emptyset\big)$ **do**

```
11        select any x ∈ {v ∈ V \ U | dist[v] < ∞} such that dist[x] is minimum
12        U ← U ∪ {x}
13        foreach y ∈ adj(x)
14            if dist[y] > dist[x] + w((x, y))
15                dist[y] ← dist[x] + w((x, y))
16                π[y] ← x
```

It is obvious that Dijkstra's algorithm terminates because at each iteration of the **while** loop (lines 10–16) precisely one vertex is added to U and since V is finite, inevitably the set $\{v \in V \setminus U \,|\, \text{dist}[v] < \infty\}$ will become $\emptyset$. Now we prove Dijkstra's algorithm computes correctly the shortest paths in G from s to all vertices.

Lemma 18. *At the termination of* DIJKSTRA, *it is the case that*

- $\forall u \in V$: *the value* $dist[u]$ *equals* $dist_G(s, u)$, *and*
- *the array* $\pi[\,]$ *represents a shortest-paths tree in* G, *rooted at* s.

Proof:
Let us first make several definitions. *A* u*-path* for any $u \in V$ is a any path from s to u. During the execution of DIJKSTRA, relative to the current value of U, for any $z \in V \setminus U$, *a* z*-special path in* G is any path $p = s, u, \ldots, w, z$, such that $|p| > 0$ and precisely one vertex in p, namely z, is not from U. *A special path* is any path that is a z-special path for some $z \in V \setminus U$. Relative to the current value of U, *the fringe* $F(U)$ is the set $\{z \in V \setminus U \,|\, \text{there exists a } z\text{-special path}\}$.
The following is a loop invariant for the **while** loop (lines 10–16):

> Every time the execution of DIJKSTRA is at line 10 the following conjunction holds:
>
> **part i:** $\forall u \in U : \text{dist}[u] = \text{dist}_G(s, u)$, and
>
> **part ii:** $\forall u \in U \setminus \{s\} : \pi[u]$ is the neighbour of u is some shortest u-path and $\pi[u] \in U$, and
>
> **part iii:** $\forall u \in V \setminus U : \text{dist}[u] < \infty$ iff $u \in F(U)$, and
>
> **part iv:** $\forall u \in F(U) : \text{dist}[u]$ is the length of a shortest u-special path and $\pi[u]$ is the neighbour of u in such a path.

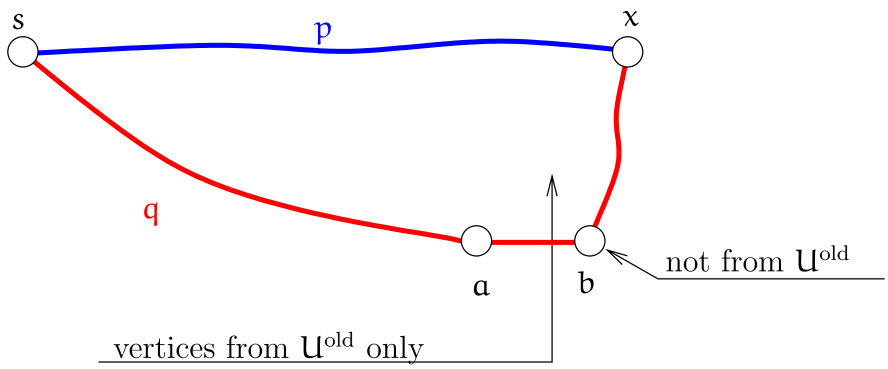
Basis. The first time the execution reaches line 10, it is the case that $U = \{s\}$ because of the assignment at line 5. **part i** holds because $\text{dist}[s] = 0$ (line 6) and $\text{dist}_G(s, s) = 0$ (by definition). **part ii** holds vacuously since $U \setminus \{s\} = \emptyset$. **part iii** holds because on the one hand $F(U) = \text{adj}(s)$ and on the other hand the assignments at lines 3 and 8 imply $\text{adj}(s)$ are the only vertices with $\text{dist}[\,] < \infty$. **part iv** holds because $\forall u \in F(U)$ the only u-special path is the edge (s, u); at lines 8 and line 9, $\text{dist}[u]$ and $\pi[u]$ are set accordingly.

Maintenance. Assume the claim holds at a certain execution of line 10 and the **while** loop is to be executed at least once more. Let us call the set U at that moment, U^{old} and after the assignment at line 12, U^{new}. So, $\{x\} = U^{\text{new}} \setminus U^{\text{old}}$. We first prove **part i** and **part ii**. We do that by considering U^{old} and x separately. We claim that for all vertices in U^{old}, their $\text{dist}[\,]$ and $\pi[\,]$ values do not change during the current iteration of the **while**

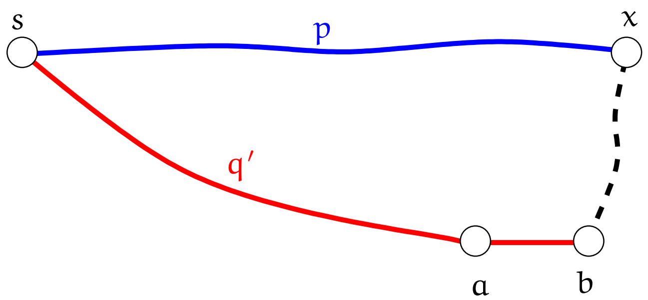
loop. But that follows trivially from the fact that by **part i** of the inductive hypothesis their dist[] values are optimal and the fact that, if the **while** loop changes the dist[] and π[] values of any vertex, that implies decreasing its dist[] value.

Consider vertex x. Before the assignment at line 12, by **part iii** of the inductive hypothesis x is a fringe vertex and so by **part iv**, its dist[] value is the length of a shortest x-special—with respect to U^{old}—path p and $\pi[x]$ is the neighbour of u in p. Now we prove that p is a shortest x-path in G.

Assume the contrary. Then there exists an x-path q that is shorter than p. Since one endpoint of q, namely s, is from U^{old}, and the other endpoint x is not from U^{old}, there is at least one pair of neighbour vertices in q such that one is from U^{old} and the other one is not from U^{old}. Among all such pairs, consider the pair a, b that is closest to s in the sense that between s and a inclusive there are only vertices from U^{old} and b is the first vertex (in direction away from s) not from U^{old}. Note that $b \neq x$, for if b were x then q would be an x-special—with respect to U^{old}—path shorter than p, and by **part iv** that is not possible. Let the subpath of q between s and b be called q'. Note that $|q'| < |q|$. By assumption, $|q| < |p|$, therefore $|q'| < |p|$. Then note q' is a special path with respect to U^{old} and $b \in F(U^{old})$. By **part iv**, dist[b] is at most $|q'|$ at the beginning of the current iteration of the **while**-loop, thus dist[b] < dist[x] and DIJKSTRA would have selected b rather than x at line 11. This contradiction refutes the assumption there exists any x-path shorter than p. So, dist[x] indeed equals $\text{dist}_G(s, x)$, therefore **part i** and **part ii** hold the next time the execution reaches line 10. The following two figures illustrate the contradiction we just derived. Initially we assumed the existence of a path q shorter than p and defined its rightmost neighbour pair a, b such that b is the vertex closest to s and not from U^{old}:



Then we concluded the subpath q' between s and b must be special with respect to U^{old} and, furthermore, shorter than p:



Immediately we concluded the algorithm whould have picked b rather than x.

It remains to prove that **part iii** and **part iv** hold after the current iteration. Have in mind that x was a fringe vertex at the beginning of the current iteration of the **while** loop but at the end of it x is in U, we exclude x from consideration. The proof of **part iii** is straightforward. As just said, x is no longer in $V \setminus U$.

- In one direction, partition the remaining vertices of $V \setminus U$ into those whose dist[] value was $< \infty$ at the beginning of the current iteration and those whose dist[] value was equal to ∞ at the at the beginning of the current iteration. By **part iii** of the inductive hypothesis, the former set were neighbours of vertices from U^{old}, so at the end of the iteration they are neighbours of vertices from U^{new} which makes them fringe vertices. On the other hand, the latter set are neighbours to x, which makes them fringe vertices with respect to U^{new}.

- In the other direction, consider the vertices from $V \setminus U$ whose dist[] value is equal to ∞ at the end of the current iteration. They can neither be neighbours to vertices from U^{old}, otherwise they would have dist[] value $< \infty$ at the beginning of the current iteration of the **while**-loop, nor can they be neighbours of x, otherwise their dist[] values would be set to some positive reals by **for**-loop at lines 13–16. Therefore, they are not fringe vertices at the end of the current iteration of the **while**-loop.

To prove **part iv**, partition $F(U^{\text{new}})$ into $F(U^{\text{old}}) \setminus \{x\}$ and $F(U^{\text{new}}) \setminus (F(U^{\text{old}}) \setminus \{x\})$—the vertices added during the current iteration of the **while**-loop. It is obvious that for every vertex u from $F(U^{\text{new}}) \setminus (F(U^{\text{old}}) \setminus \{x\})$ its only neighbour from U^{new} is x—otherwise, dist[u] would not be ∞ at the beginning of the current iteration of the **while**-loop. Then every shortest u-special path p is such that the path neighbour of u is x and so $|p|$ equals $\text{dist}_G(s, x) + w((x, u))$. We already showed that $\text{dist}_G(s, x)$ equals dist[x] during the current iteration of the **while**-loop. We note that at line 16, dist[u] is assigned precisely dist[x] + $w((x, u))$. Clearly, **part iv** holds for u.

Now consider any vertex u in $F(U^{\text{old}}) \setminus \{x\}$. If dist[u] is not changed, in other words decreased, during the current iteration of the **while**-loop, **part iv** holds by the induction hypothesis regardless of whether u is or is not a neighbour of x. Suppose dist[u] is decreased during the current iteration of the **while**-loop. u must be a neighbour of x because the only place dist[u] can be altered is at line 15 that is executed within the **for**-loop (lines 13–16). However, u is a neighbour of at least one vertex from U^{old}, otherwise u would not be a vertex from $F(U^{\text{old}})$. By **part iv** of the induction hypothesis, at the beginning of the current iteration of the **while**-loop, it is the case that dist[u] = $|p|$, where p is a shortest u-special path with respect to U^{old}. The fact that dist[u] was altered at line 15 means that

$$\text{dist}[x] + w((x, u)) < |p|$$

It follows there is a u-special path q with respect to U^{new} such that the path neighbour of u in q is x, and $|q| = \text{dist}[x] + w((x, u))$. Furthermore, $|q|$ is the minimum length of any u-special path q with respect to U^{new} such that the path neighbour of u in q is x because $\text{dist}[x] = \text{dist}_G(s, x)$ as we already proved. There cannot be a shorter than q, u-special path with respect to U^{new}—assuming the opposite leads to a contradiction because the path neighbour of u in that alleged path cannot be x and cannot be any other vertex from U^{new}. That concludes the proof of **part iv**.

Termination. Consider the moment when the execution is at line 10 for the last time. It is either the case that $U = V$, or $U \subset V$ but all vertices in $V \setminus U$ have dist[] values equal to ∞. In the former case, the claim of this lemma follows directly from **part i** and **part ii** of the invariant. In the latter case, it is easy to see that every vertex w such that $\text{dist}[w] = \infty$ is such that no path exists from s to w—assuming the opposite leads to a contradiction because that alleged path must have neighbour vertices a and b, such that $\text{dist}[a] < \infty$, $\text{dist}[b] = \infty$, and $(a, b) \in E(G)$; clearly, b would have gotten finite dist[] value as a neighbour of a during the iteration of the **while**-loop when the value of the x variable at line 11 was a. It follows that DIJKSTRA assigns ∞ precisely to the dist[] of those vertices that are not reachable from s, and all other ones are dealt with correctly. $\square$