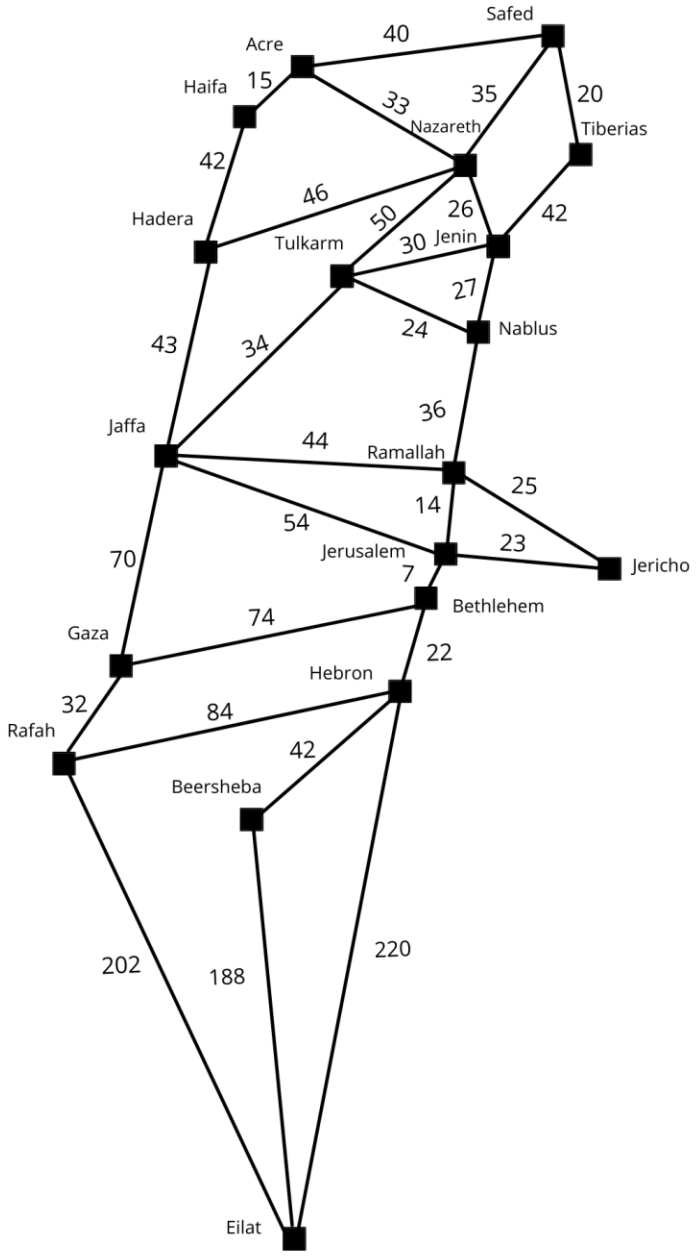


BEST-FIRST SEARCH (BEST-FS)

- The search space is evaluated according to a heuristic function.
- Nodes yet to be evaluated are kept on an OPEN list(represented as a priority queue)
- Nodes that have already been evaluated are stored on a CLOSED list.
- The OPEN list is then built in **order of f(n).**

This makes best-first search fundamentally greedy because it always chooses the best local opportunity in the search frontier.

dr. Mohamed Khalil
Palestine Technical University

Frontier

- Best-first search
- Idea: use an evaluation function f(n) for each node
- f(n) provides an estimate for the total cost.
- Expand the node n with smallest f(n).
- Implementation: Order the nodes in fringe increasing order of cost.
- Special cases:
  - greedy best-first search
  - A* search

Frontier

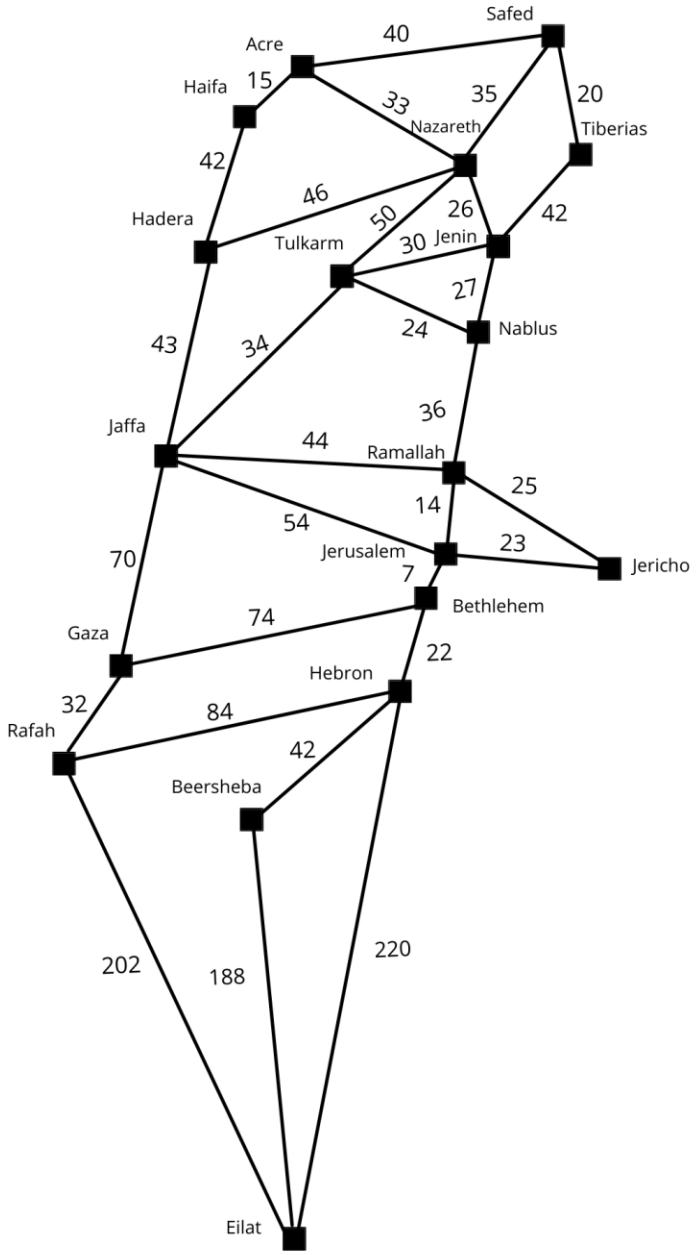
- The search frontier is defined as the set of node opportunities that can be searched next.
- In Best-First search, the frontier is a priority queue sorted in f(n) order.
- Given the strict order of f(n), the selection of the node to evaluate from the priority queue is greedy

Properties of best-first search

- Complete? No – can get stuck in loops.
- Time? $O(b^m)$, but a good heuristic can give dramatic improvement
- Space? $O(b^m)$ -- keeps all nodes in memory
- Optimal? No

a solution can be found in a longer path (higher h(n) with a lower g(n) value.

Palestine with step costs in km



Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Best-first search

- Evaluation function f(n) = h(n) (heuristic)
 = estimate of cost from n to goal

- e.g., $h_{SLD}(n)$ = straight-line distance from n to Bucharest

- Greedy best-first search expands the node that appears to be closest to goal

Greedy Best-first search example

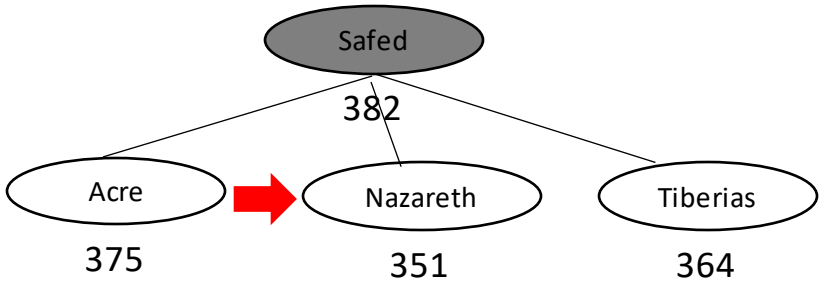
382

Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example

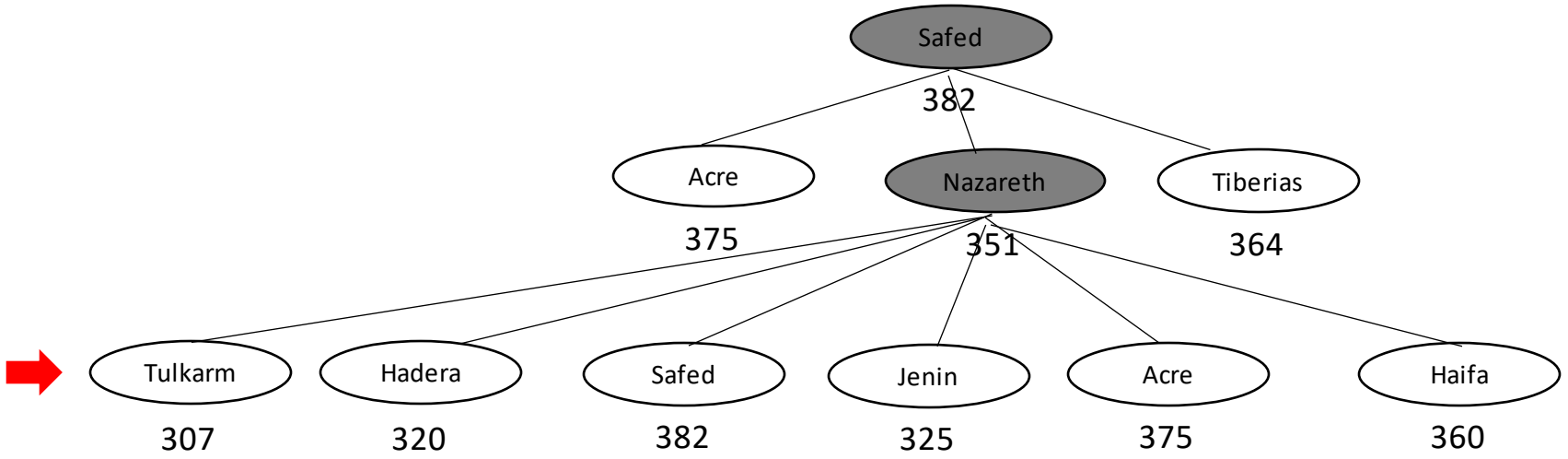


Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example

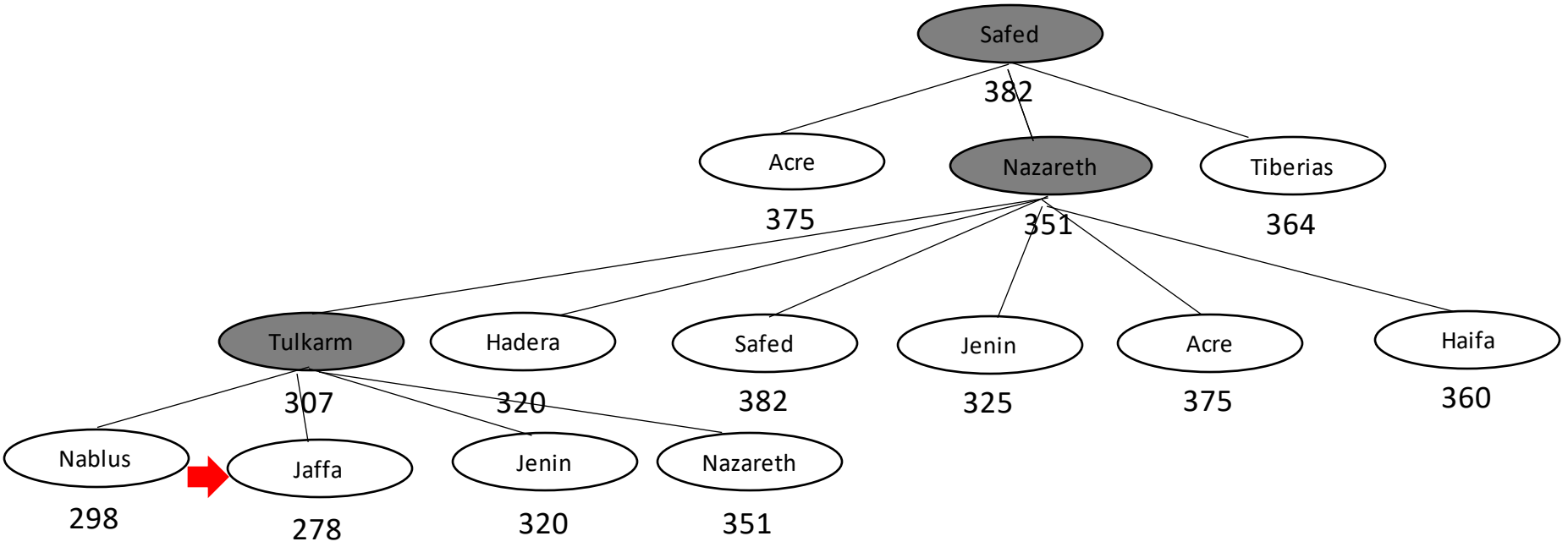


Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example

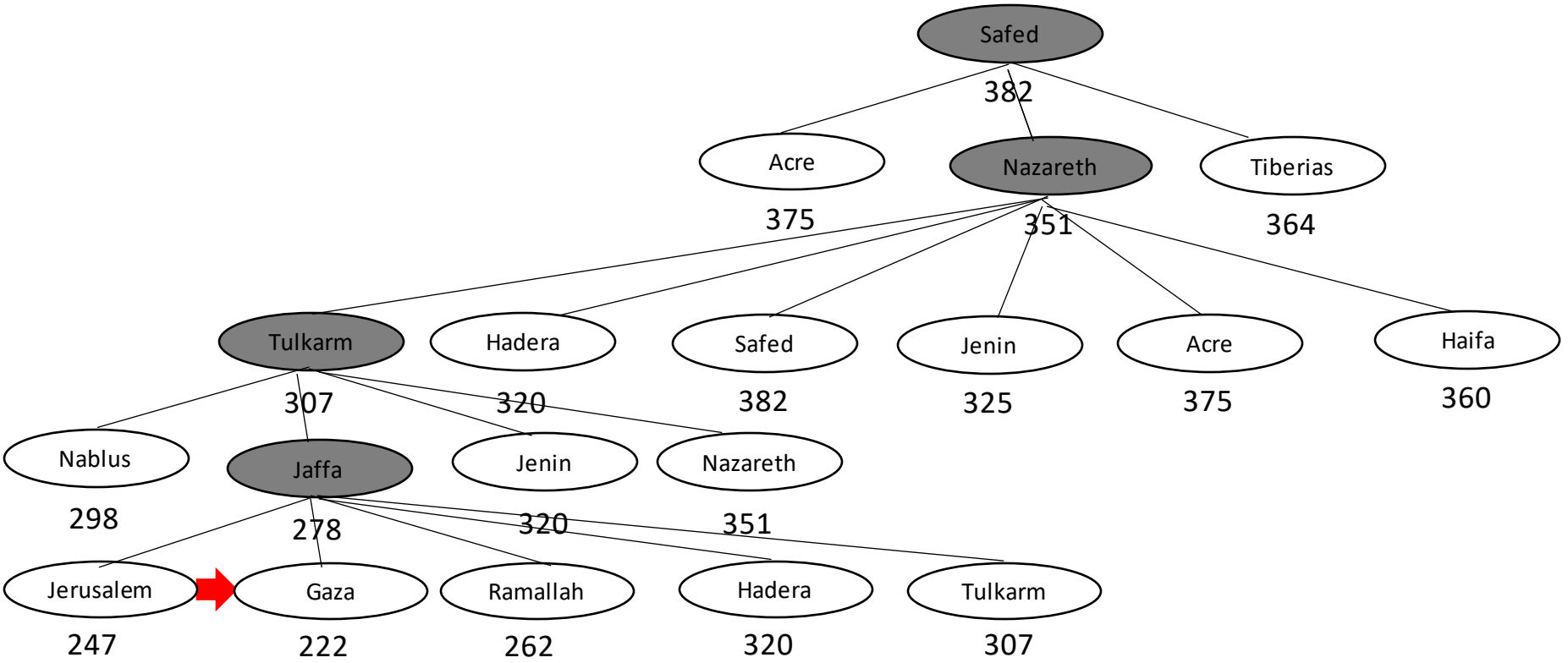


Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example

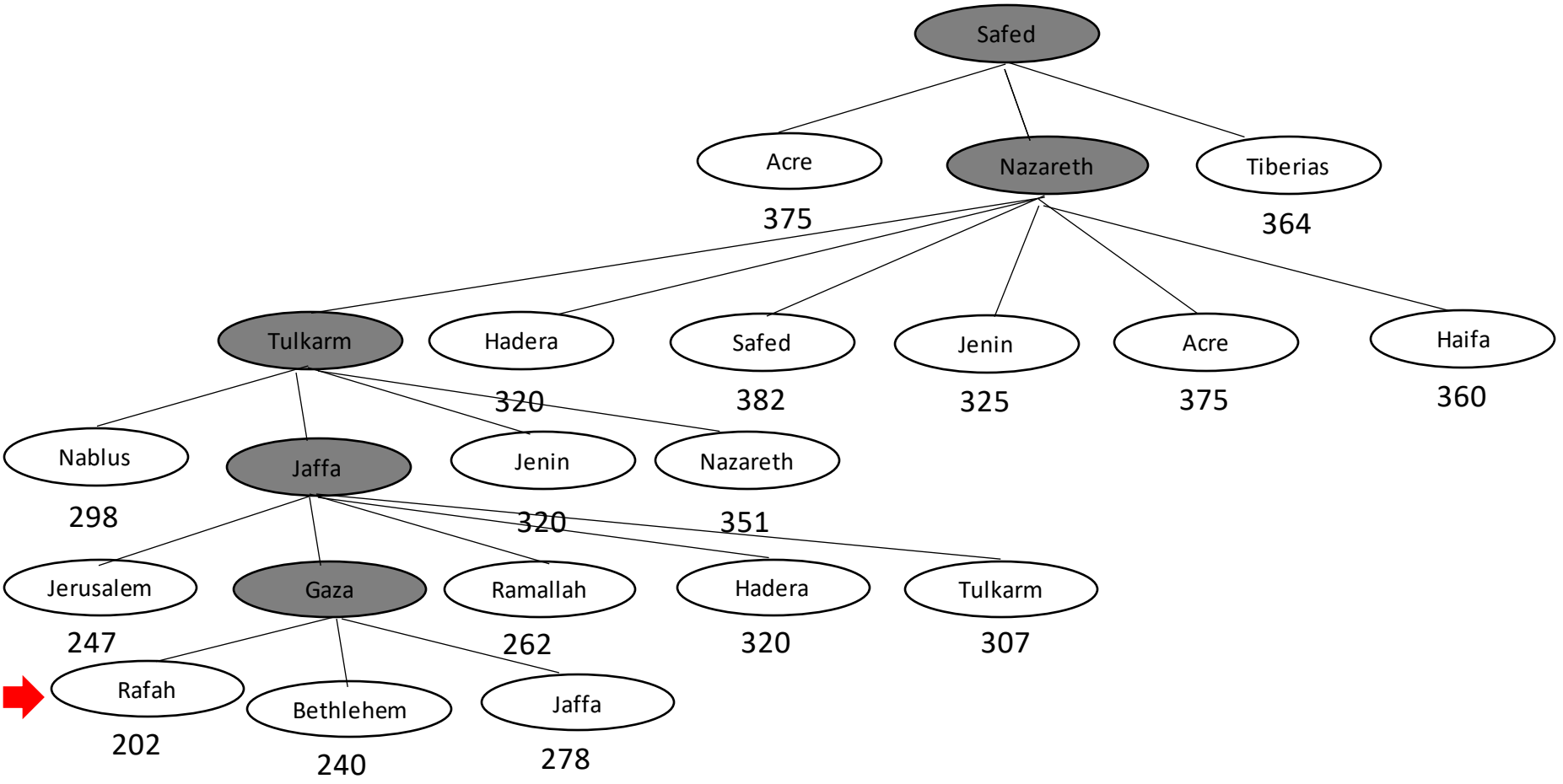


Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example

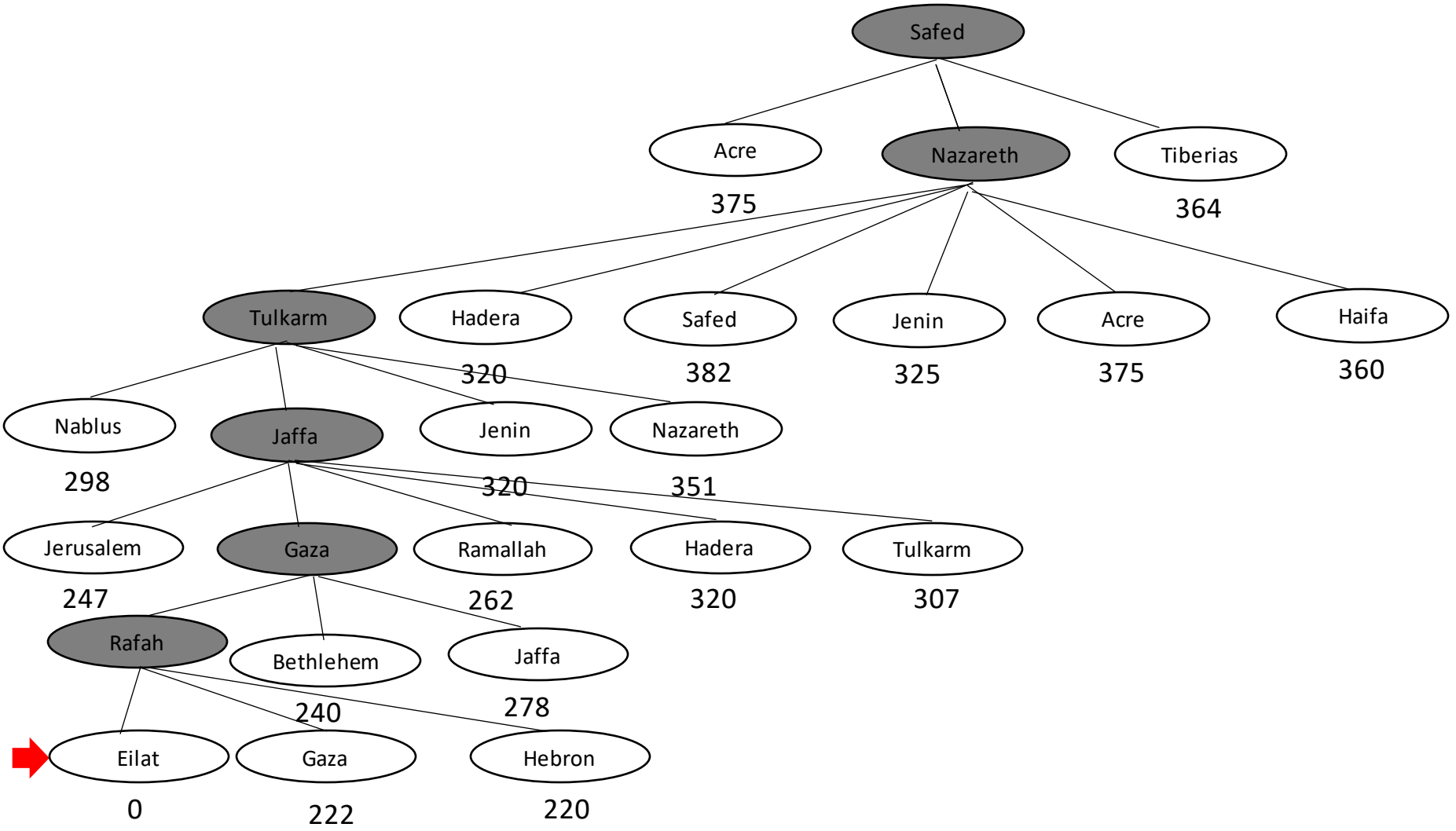


Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Greedy Best-first search example



Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

A * search

- Idea: avoid expanding paths that are already expensive
- Evaluation function f(n) = g(n) + h(n)
- g(n) = cost so far to reach n
- h(n) = estimated cost from n to goal
- f(n) = estimated total cost of path through n to goal

A*SEARCH

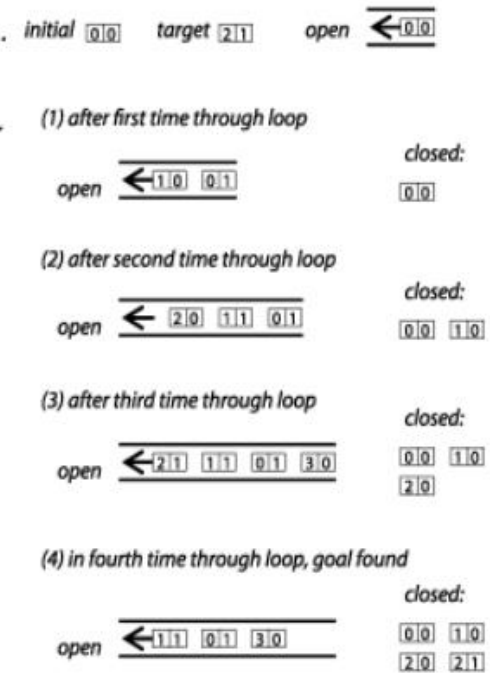
Best	Average	Worst
O(b*d)	O(b^d)	O(b^d)

Priority Queue

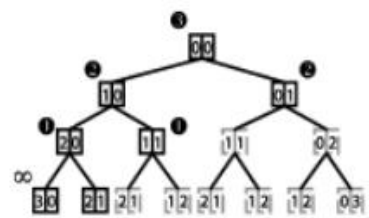
Heuristics

Set

```
search  (initial, goal)
1.    initial.depth = 0
2.    open = new PriorityQueue
3.    closed = new Set
4.    insert (open, copy(initial))
5.    while (open is not empty) do
6.      n = minimum (open)
7.      insert (closed, n)
8.      if (n = goal) then return "Solution"
9.      foreach valid move m at n do
10.         next = state when playing m at n
11.         next.depth = n.depth + 1
12.         if (closed contains next) then
13.             prior = state in closed matching next
14.             if (next.score < prior.score) then
15.                 remove (closed, prior)
16.                 insert (open, next)
17.         else
18.             insert (open, next)
19.   return "No Solution"
end
```



closed
explored
open
unexplored



Admissible heuristic

- a heuristic function is said to be **admissible** if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path.

n is a node

h is a heuristic

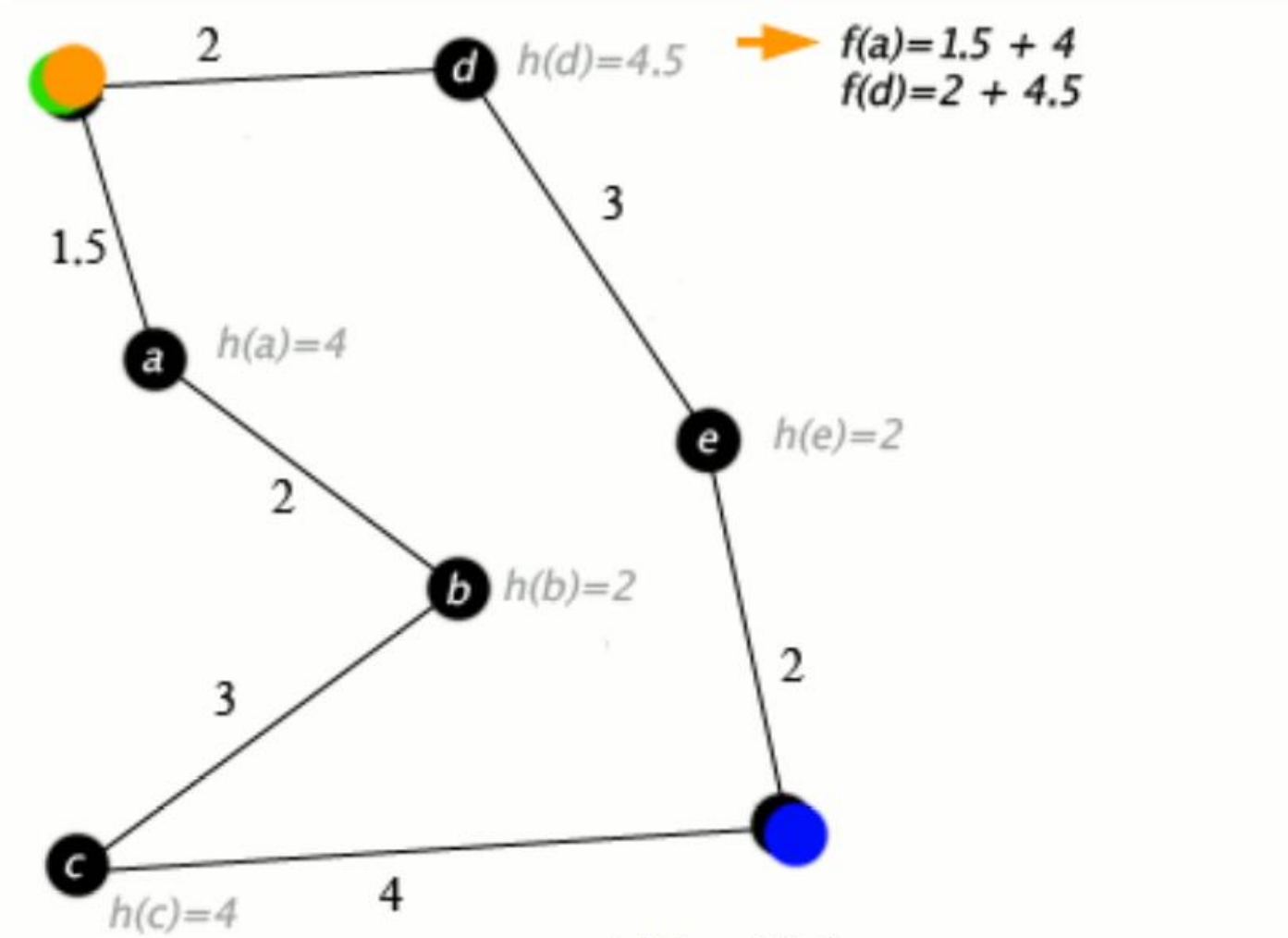
$h(n)$ is cost indicated by h to reach a goal from n

$h^*(n)$ is the actual cost to reach a goal from n

$h(n)$ is admissible if, $\forall n$

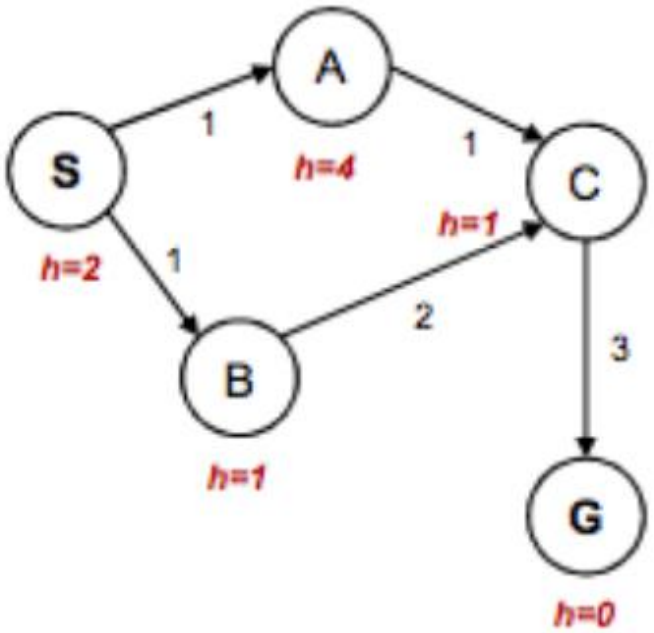
$$h(n) \leq h^*(n)$$

A* search

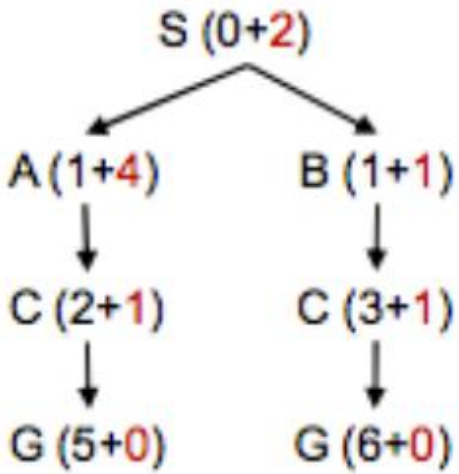


A * search

State space graph

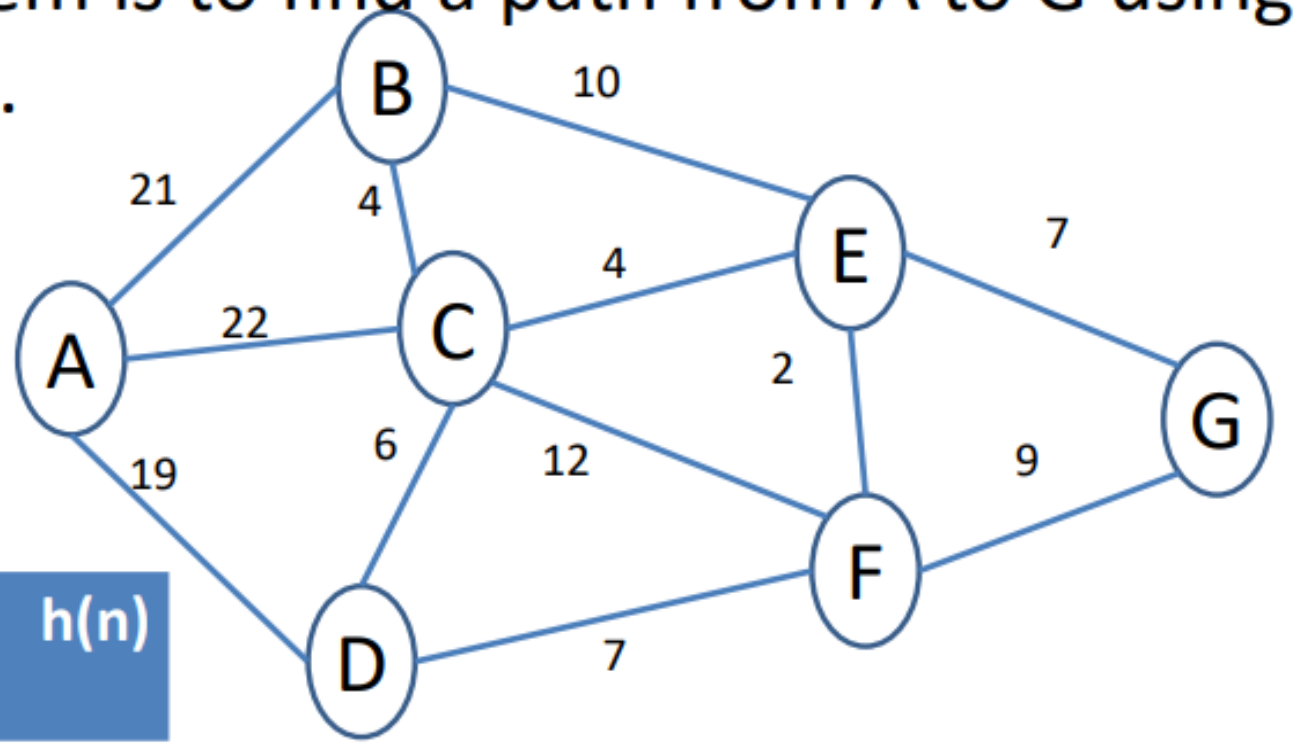


Search tree



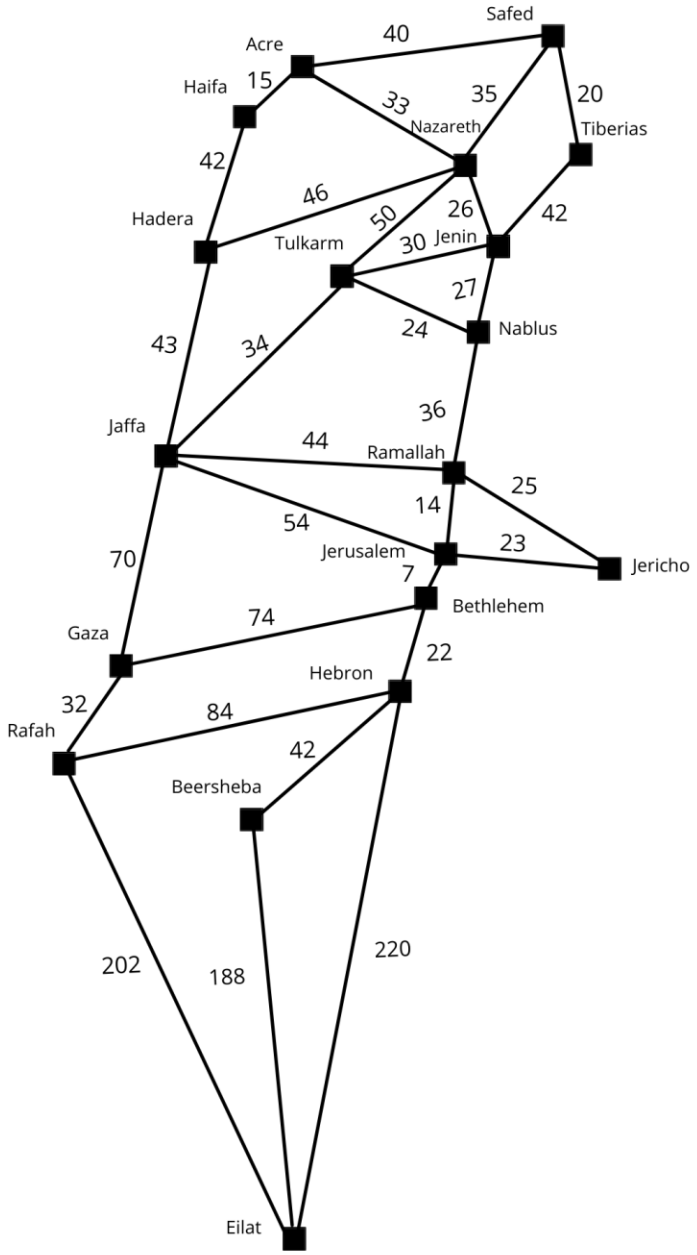
dr. Mohamed Khalil
Palestine Technical University

the problem is to find a path from A to G using A* algorithm.



nodes	h(n)
A	34
B	19
C	14
D	18
E	7
F	9
G	0

Palestine with step costs in km



Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

dr. Mohamed Khalil
Palestine Technical University

Open List: Safed

A * search Example

0+382=382

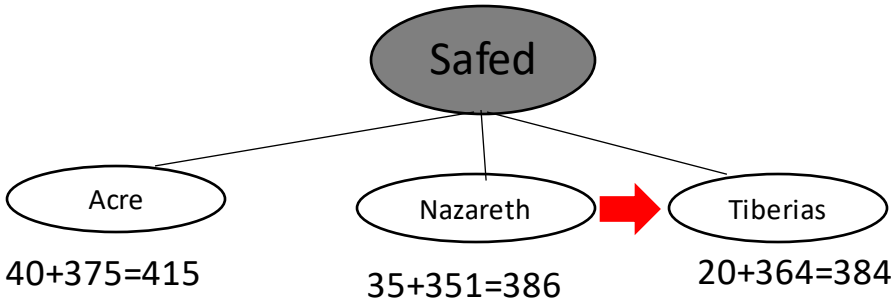
Straight-line Distance to Eilat

Safed	382
Eilat	0
Acre	375
Nazareth	351
Tiberias	364
Tulkarm	307
Haifa	360
Hadera	320
Nablus	298
Ramallah	262
Jerusalem	247
Bethlehem	240
Jericho	260
Jenin	325
Jaffa	278
Hebron	220
Rafah	202
Gaza	222
Beersheba	189

We start with our initial state Safed.
We make a node and add it to the open list. Since it's the only thing on the open list,
we expand the node. Think of the open list as a priority queue (or heap) that sorts the nodes inside of it according to their g()+h() score

Open List:
Tiberias
Nazareth
Acre

A * search Example



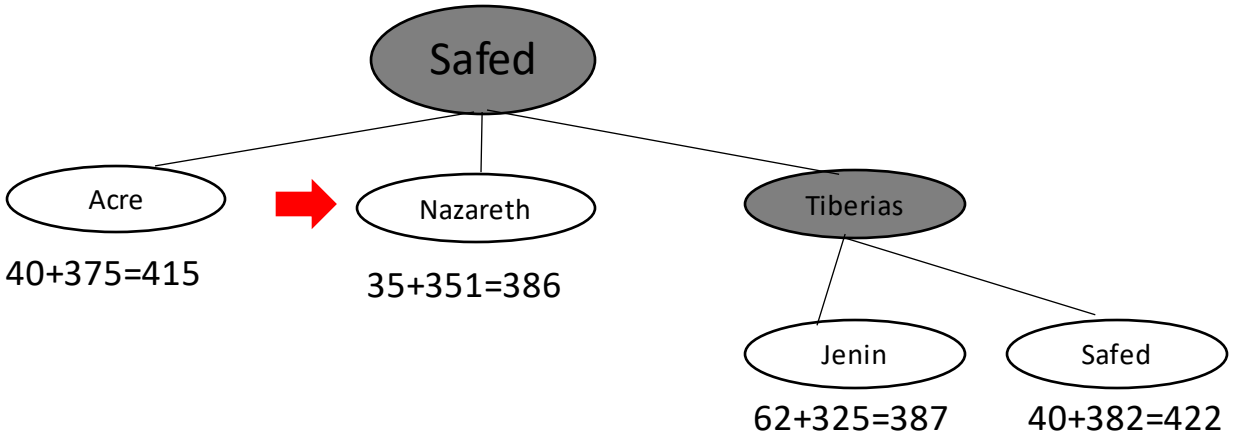
Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

We add the three nodes we found to the open list. We sort them according to the g()+h() calculation.

Open List:
Nazareth
Jenin
Acre

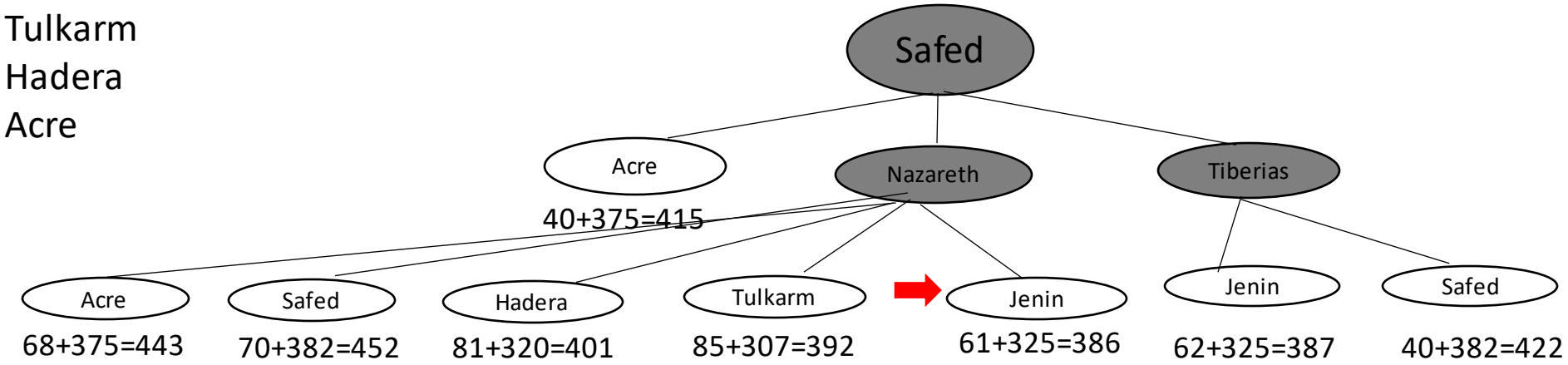
A * search Example



When we expand Tiberias, we run into Safed again. But we've already expanded this node once; so, we don't add it to the open list again.
We see that Nazareth is at the top of the open list; so, it's the next node we will expand.

A * search Example

Open List:
Jenin
Tulkarm
Hadera
Acre



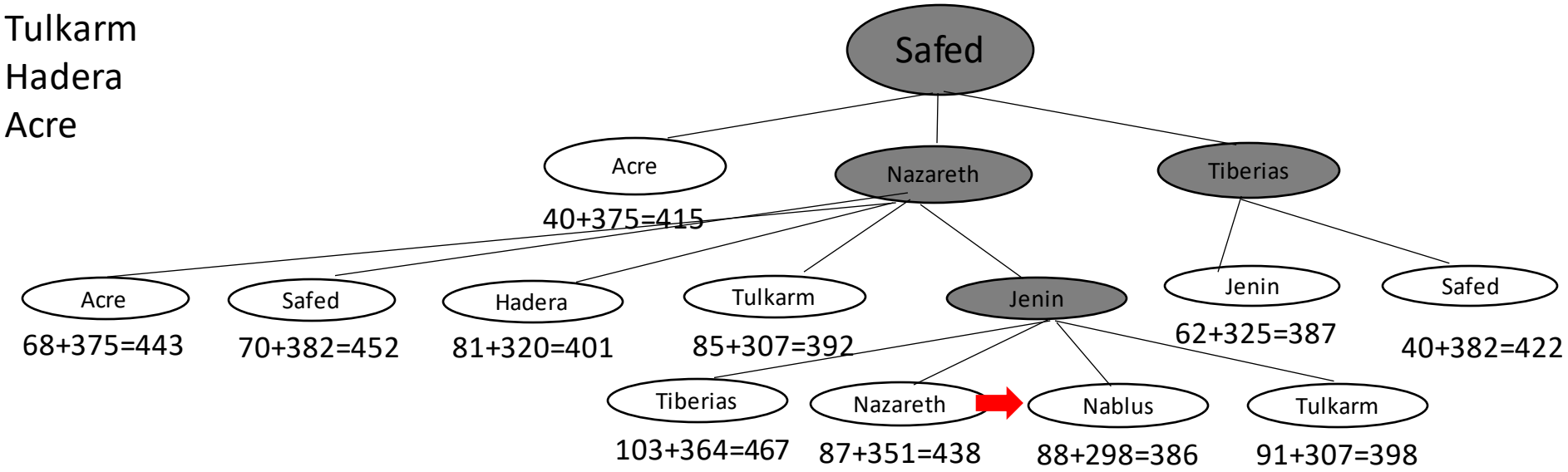
Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

When we expand Nazareth, we run into Acre and Safed again. But we've already expanded those nodes once; so, we don't add it to the open list again.

A * search Example

Open List:
Nablus
Tulkarm
Hadera
Acre



Straight-line Distance to Eilat

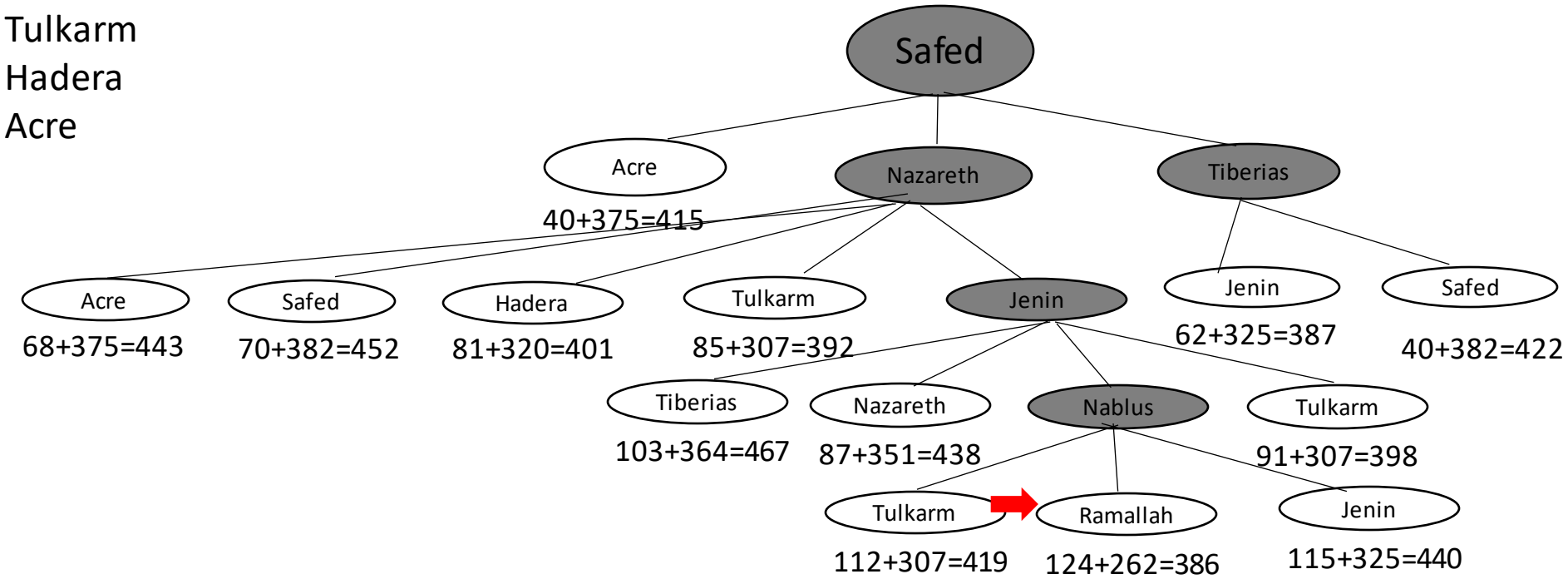
Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

When we expand Jenin, we run into Nazareth and Tiberias again.
But we've already expanded those nodes once; so, we don't add them to the open list again.
Nablus will be the next node we should expand – it's at the top of the sorted open list.

dr. Mohamed Khalil
Palestine Technical University

A * search Example

Open List:
Ramallah
Tulkarm
Hadera
Acre



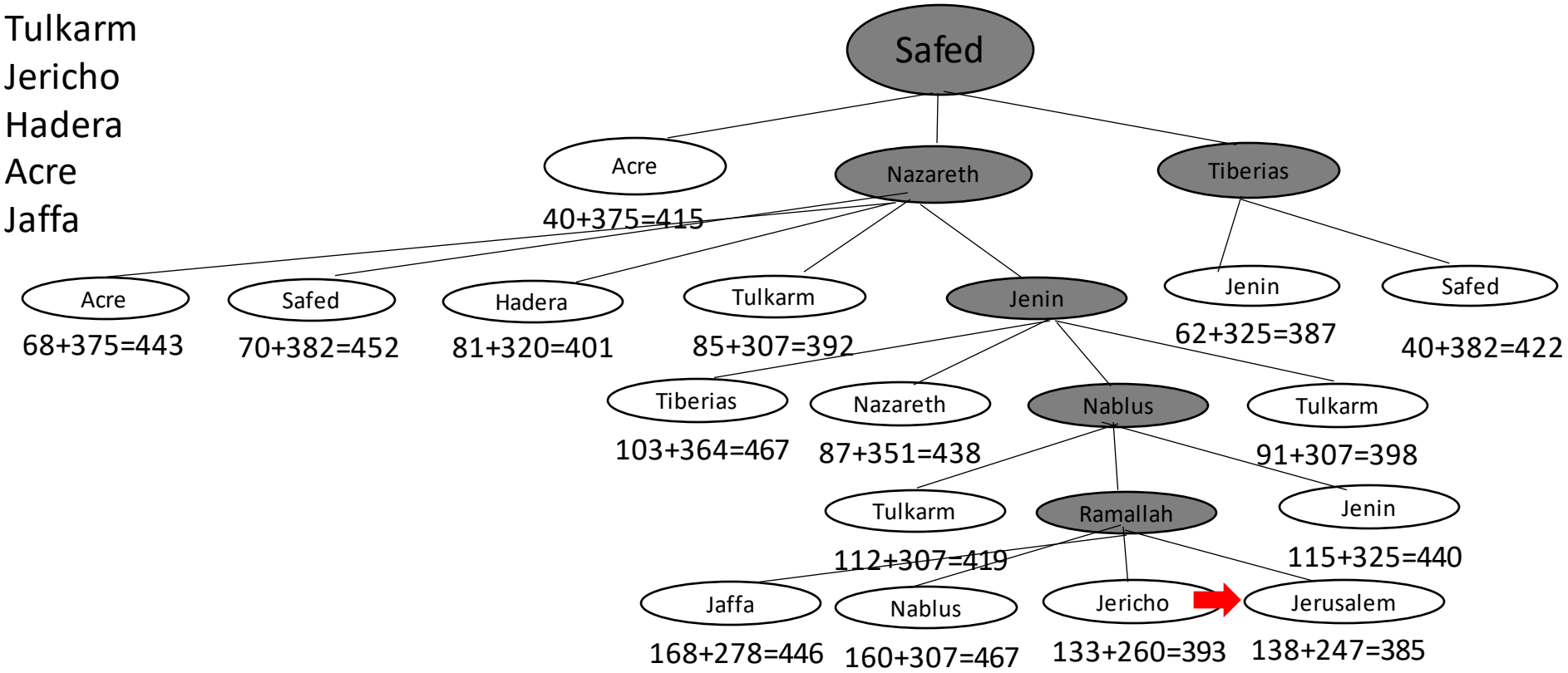
Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

When we expand Nablus, we run into Tulkarm and Jenin again.
But we've already expanded Jenin node once; so, we don't add it to the open list again
Tulkarm Remains in it's old location in the open list
Ramallah will be the next node we should expand – it's at the top of the sorted open list.

A * search Example

Open List:
Jerusalem
Tulkarm
Jericho
Hadera
Acre
Jaffa



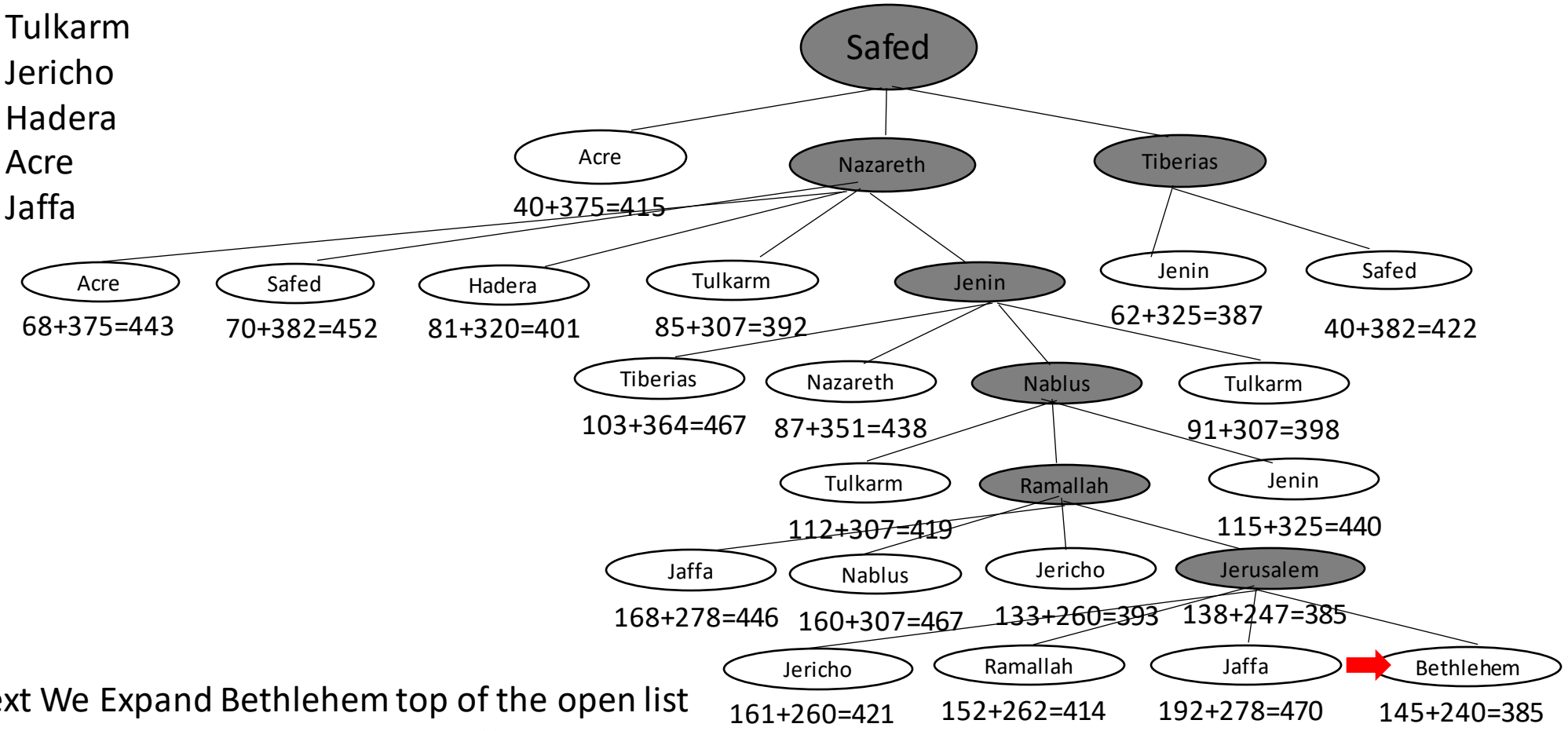
Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

Next We Expand Jerusalem It has the lowest f(n) (top of the open list)

A * search Example

Open List:
Bethlehem
Tulkarm
Jericho
Hadera
Acre
Jaffa



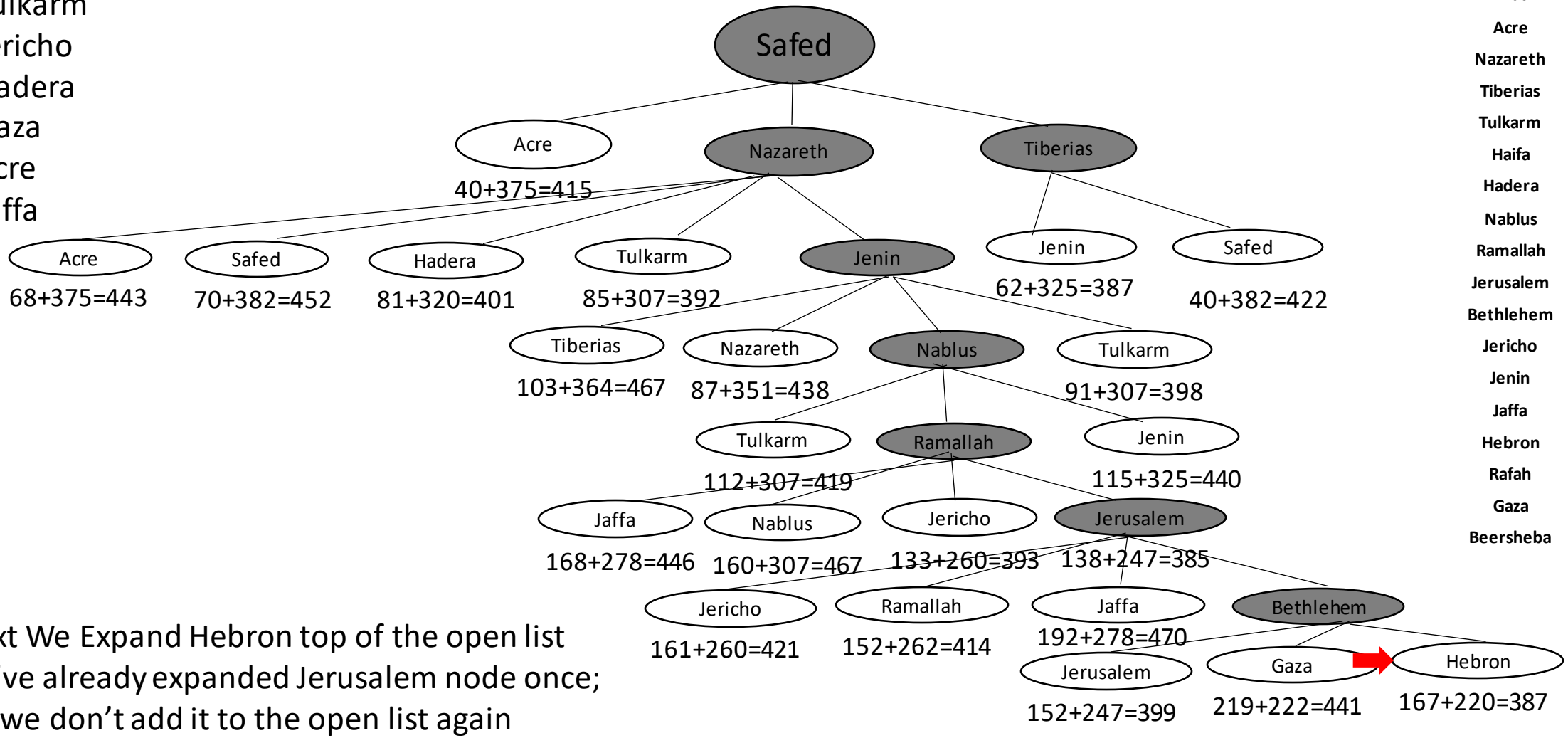
Straight-line Distance to Eilat

Safed	**382**
Eilat	**0**
Acre	**375**
Nazareth	**351**
Tiberias	**364**
Tulkarm	**307**
Haifa	**360**
Hadera	**320**
Nablus	**298**
Ramallah	**262**
Jerusalem	**247**
Bethlehem	**240**
Jericho	**260**
Jenin	**325**
Jaffa	**278**
Hebron	**220**
Rafah	**202**
Gaza	**222**
Beersheba	**189**

Next We Expand Bethlehem top of the open list
We run into Ramallah ,Jericho , Jaffa again
We don't add them again to the list

A * search Example

Open List:
Hebron
Tulkarm
Jericho
Hadera
Gaza
Acre
Jaffa



Straight-line Distance to Eilat

Safed	382
Eilat	0
Acre	375
Nazareth	351
Tiberias	364
Tulkarm	307
Haifa	360
Hadera	320
Nablus	298
Ramallah	262
Jerusalem	247
Bethlehem	240
Jericho	260
Jenin	325
Jaffa	278
Hebron	220
Rafah	202
Gaza	222
Beersheba	189

Next We Expand Hebron top of the open list
we've already expanded Jerusalem node once;
so, we don't add it to the open list again

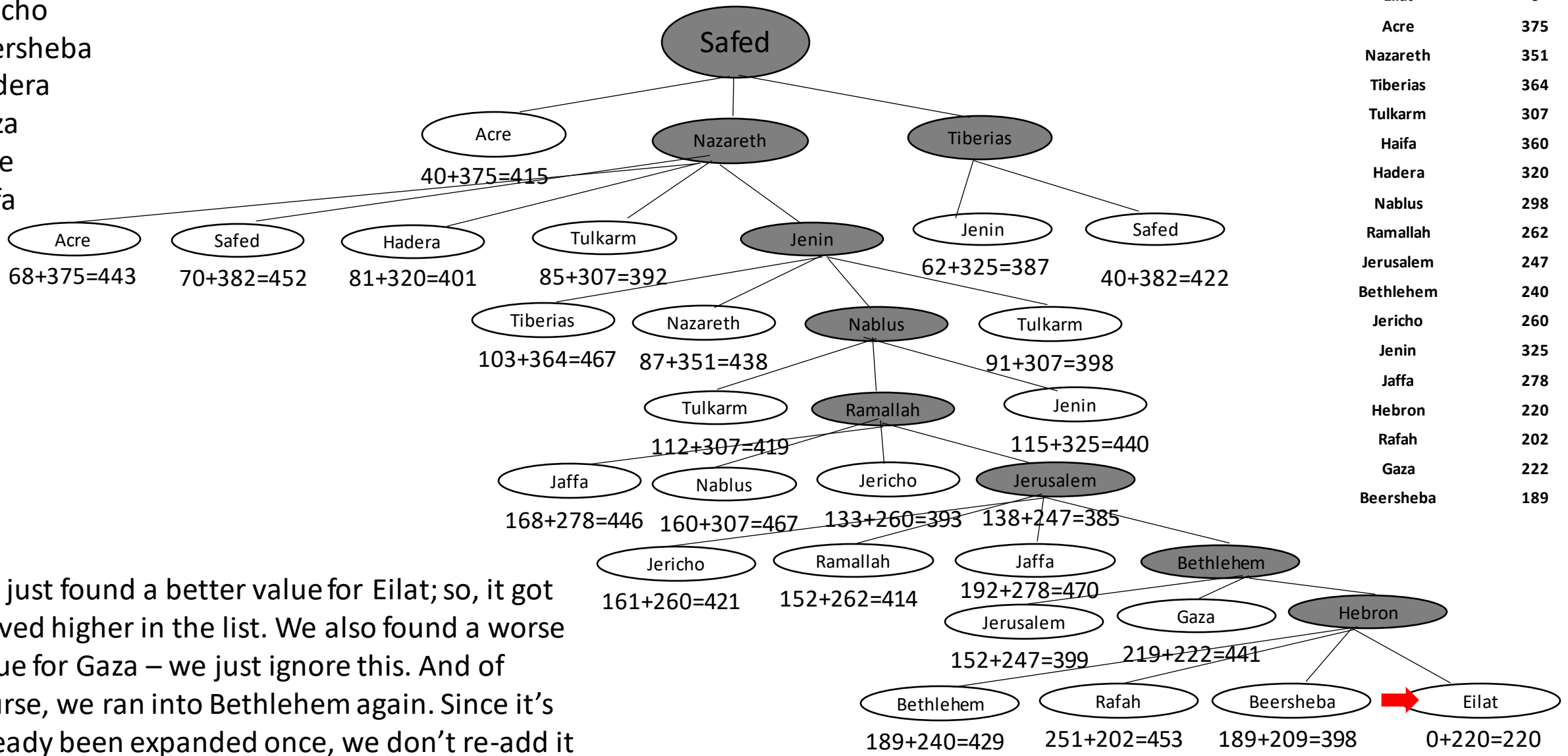
dr. Mohamed Khalil
Palestine Technical University

A * search Example

Open List:
Eilat
Tulkarm
Jericho
Beersheba
Hadera
Gaza
Acre
Jaffa

Straight-line Distance to Eilat

Safed	382
Eilat	0
Acre	375
Nazareth	351
Tiberias	364
Tulkarm	307
Haifa	360
Hadera	320
Nablus	298
Ramallah	262
Jerusalem	247
Bethlehem	240
Jericho	260
Jenin	325
Jaffa	278
Hebron	220
Rafah	202
Gaza	222
Beersheba	189



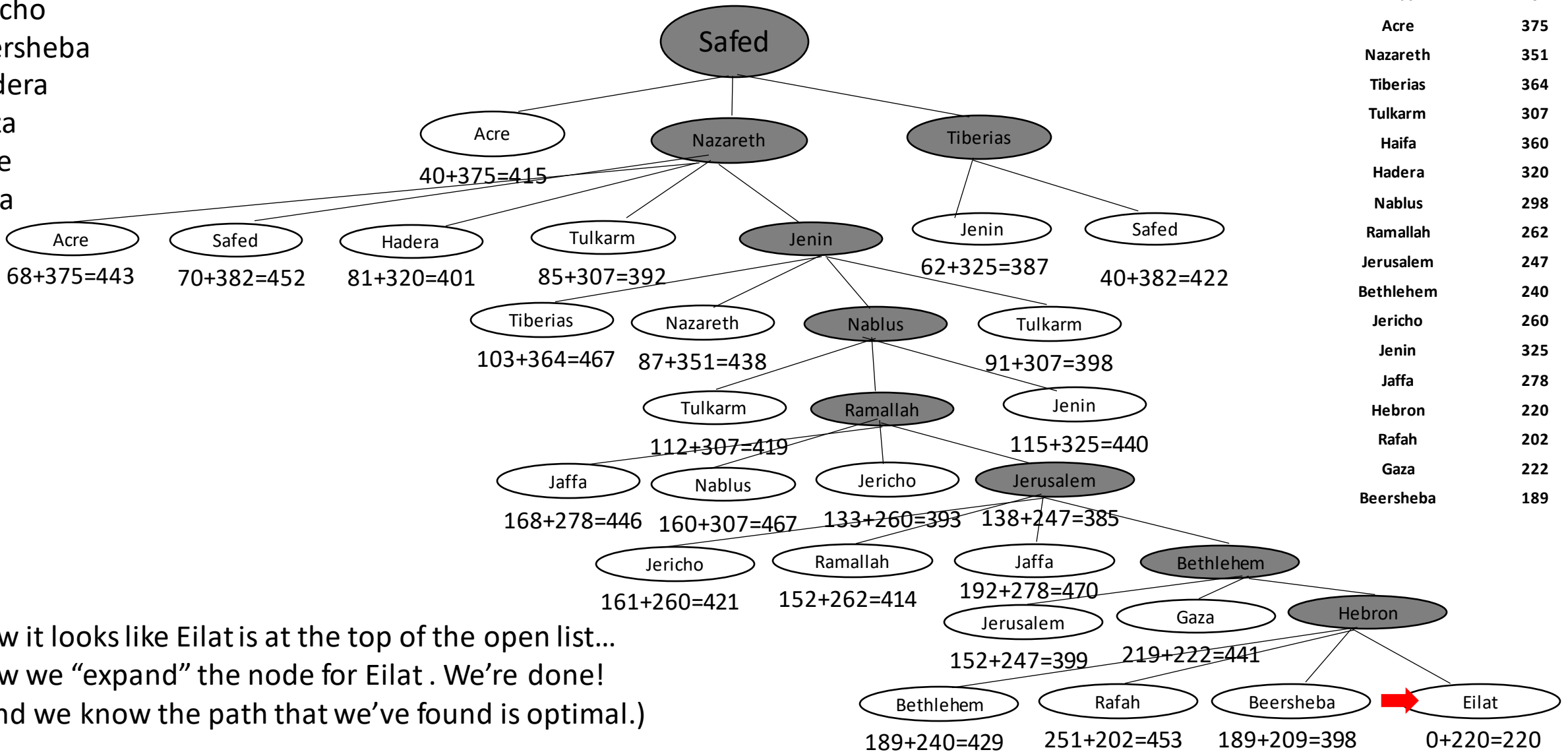
We just found a better value for Eilat; so, it got moved higher in the list. We also found a worse value for Gaza – we just ignore this. And of course, we ran into Bethlehem again. Since it's already been expanded once, we don't re-add it to the Open List.

Open List:
Eilat
Tulkarm
Jericho
Beersheba
Hadera
Gaza
Acre
Jaffa

A * search Example

Straight-line Distance to Eilat

Safed	382
Eilat	0
Acre	375
Nazareth	351
Tiberias	364
Tulkarm	307
Haifa	360
Hadera	320
Nablus	298
Ramallah	262
Jerusalem	247
Bethlehem	240
Jericho	260
Jenin	325
Jaffa	278
Hebron	220
Rafah	202
Gaza	222
Beersheba	189



Now it looks like Eilat is at the top of the open list…
Now we "expand" the node for Eilat . We're done!
(And we know the path that we've found is optimal.)

Analyzing the heuristic function

- ***f(n) = h(n)*** ***(Greedy best first search)***
- If h(n) = *constant(or zero) g=g+1* then A* as BFS
- **If h(n) = 0 for all n then *f(n) = g(n)***

  This heuristic is admissible, A*performs exactly as UCS
- **If h(n) = h*(n) for all n**
  - Only nodes on optimal solution path are expanded
  - No unnecessary work is performed ‰
- **The closer h is to h***, The fewer extra nodes that will be expanded.

Properties of Best First Search(A*)

- Complete? Yes – as long as the memory supports the depth and branching factor of the tree.

- Time? $O(b^d)$, but a good heuristic can give dramatic improvement.

- Space? $O(b^d)$ -- keeps all nodes in memory.

- Optimal? Yes

If h is admissible, A* will always find a least cost path to the goal

Resources

- Distance to
- Wikipedia
- Utc.city
- Google Maps

Palestinian Cities in Arabic

Acre-عكا
Eilat - الرشراش ام
Nablus - نابلس
Jenin - جنين
Tulakrm - طولكرم
Jerusalem - القدس

Safed - صفد
Haifa - حيفا
Tiberias - طبريا
Hadera - الخضيرة
Nazareth - الناصرة
Jaffa - يافا
Hebron - الخليل

Ramallah - رام الله
Jericho - أريحا
Bethlehem - بيت لحم
Gaza - غزة
Beersheba - بئر السبع
Rafah - رفح

By . Asad Asad