

Best-First Search

Minimizing Space or Time

IDA*

Save space, take more time

A* space complexity

- » What does the space complexity of A* depend upon?

A* space complexity – 2

- » What does the space complexity of A* depend upon?
 - > Saves all found nodes

A* space complexity – 3

- » What does the space complexity of A* depend upon?
 - > Saves all found nodes
- » What does the number of found nodes depend upon?

A* space complexity – 4

- » **What does the space complexity of A* depend upon?**
 - > **Saves all found nodes**
- » **What does the number of saved nodes depend upon?**
 - > **Depends upon the branching factor (B) and height of tree (H)**

A* space complexity – 5

» What is the space complexity of A*?

A* space complexity – 6

» What is the space complexity of A*?

> Approximately B^H

Space saving

» **How can we save space?**

Space saving – 2

- » **How can we save space?**
 - > **Not keep all the found nodes**

Space saving – 3

- » **How can we save space?**
 - > **Not keep all the found nodes**
- » **Which ones do we keep?**

Space saving – 4

- » **How can we save space?**
 - > **Not keep all the found nodes**
- » **Which ones do we keep?**
 - > **The ones in the current path**

Space saving – 5

- » **How can we save space?**
 - > **Not keep all the found nodes**
- » **Which ones do we keep?**
 - > **The ones in the current path**
- » **How do we get the nodes we threw away?**

Space saving – 6

- » **How can we save space?**
 - > **Not keep all the found nodes**
- » **Which ones do we keep?**
 - > **The ones in the current path**
- » **How do we get the nodes we threw away?**
 - > **By regenerating them when a different path is to be extended**

Iterative deepening

» **How does iterative deepening work?**

Iterative deepening – 2

- » **How does iterative deepening work?**
 - > **By doing depth-first search with increasing depth**

Iterative deepening – 3

- » **How does iterative deepening work?**
 - > **By repeating depth-first search with increasing depth**
- » **What can we use instead of depth?**

Iterative deepening – 4

- » **How does iterative deepening work?**
 - > **By repeating depth-first search with increasing depth**
- » **What can we use instead of depth? What analogous feature is the A* algorithm based on?**

Iterative deepening – 5

- » **How does iterative deepening work?**
 - > **By repeating depth-first search with increasing depth**
- » **What can we use instead of depth? What analogous feature is the A* algorithm based on?**
 - > **The $f(N)$ function**

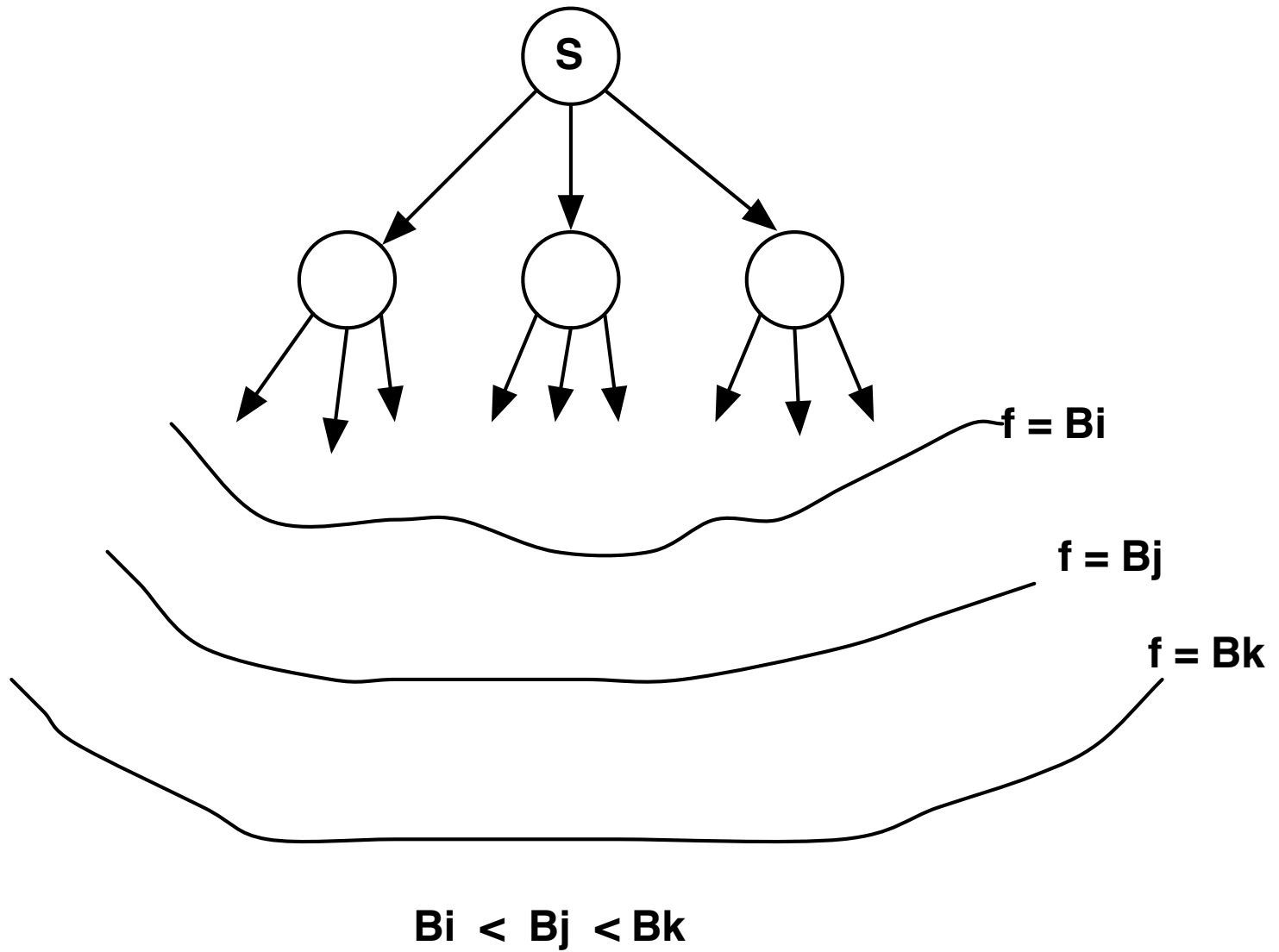
Iterative deepening – 6

- » **How does iterative deepening work?**
 - > **By repeating depth-first search with increasing depth**
- » **What can we use instead of depth? What analogous feature is the A* algorithm based on?**
 - > **The $f(N)$ function**
- » **How do we use the $f(N)$ function?**

Iterative deepening – 7

- » **How does iterative deepening work?**
 - > **By repeating depth-first search with increasing depth**
- » **What can we use instead of depth? What analogous feature is the A* algorithm based on?**
 - > **The $f(N)$ function**
- » **How do we use the $f(N)$ function?**
 - > **Do depth-first search with increasing f-limit**

A view of iterative $f(N)$ deepening



IDA* algorithm

bound = f(start_node)
found = false

while not found do

depth-first search from start_node for nodes N
such that $f(N) \leq \text{bound}$

if goal_found

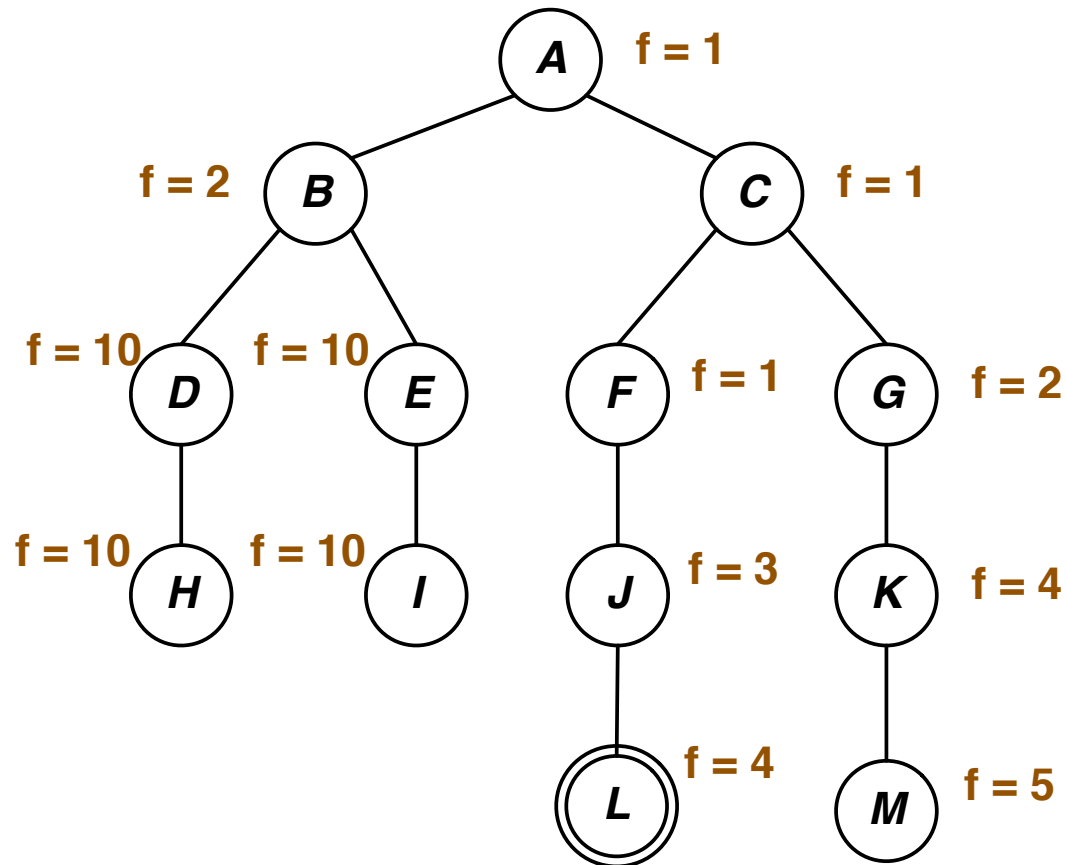
then found \leftarrow true

else bound = min { $f(N) \mid N$ generated by search $\cdot f(N) > \text{bound}$ }
fi

end

Exercise question

Trace the execution of A* for the tree. How many nodes are generated by A* and IDA*? Count all re-generated nodes.



IDA* performance

- » **What would we examine in thinking about IDA* performance?**

IDA* performance – 2

- » **What would we examine in thinking about IDA* performance?**
 - > **Space**
 - > **Time**

IDA* space performance

- » **Space is not a consideration, why?**

IDA* space performance – 2

- » **Space is not a consideration, why?**
 - > **Only one path is kept at a time**

IDA* time performance

◇ Need to look at time.

» **What is the problem if only one path is kept at any time?**

IDA* time performance – 2

- ◇ Need to look at time.
 - » **What is the problem if only one path is kept at any time?**
 - > **Have to regenerate paths that are to be extended**

IDA* regeneration performance

- » Under what conditions is the overhead of re-generating nodes
 - > High?

IDA* regeneration performance – 2

- » Under what conditions is the overhead of re-generating nodes
 - > High?
 - When there are many different f values

IDA* regeneration performance – 3

» **Under what conditions is the overhead of re-generating nodes**

> **High?**

- **When there are many different f values**
- **Extreme case have one new node per path regenerated**

IDA* regeneration performance – 4

» **Under what conditions is the overhead of re-generating nodes**

> **High?**

- **When there are many different f values**
- **Extreme case have one new node per path regenerated**
- **Unacceptable overhead**

IDA* regeneration performance – 5

- » Under what conditions is the overhead of re-generating nodes
 - > Low?

IDA* regeneration performance – 5

- » Under what conditions is the overhead of re-generating nodes
 - > Low?
 - When there are equal f values

IDA* regeneration performance – 6

- » **Under what conditions is the overhead of re-generating nodes**
 - > **Low?**
 - **When there are equal f values**
 - **Each path generates many new nodes**

IDA* regeneration performance – 7

» **Under what conditions is the overhead of re-generating nodes**

> **Low?**

- **When there are equal f values**
- **Each path generates many new nodes**
- **Regenerated nodes are a small fraction of total generated nodes**

Monotonic function

» **What does monotonic function mean?**

Monotonic function

- » **What does monotonic function mean?**
 - > **A function that is either entirely non-increasing or non-decreasing**

f function monotonicity

- » For the A* algorithm does it matter if the f function is non-monotonic?

f function monotonicity – 2

» For the A* algorithm does it matter if the f function is non-monotonic?

> No

f function monotonicity – 3

- » For the A* algorithm does it matter if the f function is non-monotonic?
 - > No
- » Why?

f function monotonicity – 4

- » **For the A* algorithm does it matter if the f function is non-monotonic?**
 - > **No**
- » **Why?**
 - > **The A* algorithm has all the paths and will always expand the best one first**

f function monotonicity – 5

- » For the IDA* algorithm does it matter if the f function is non-monotonic?

f function monotonicity – 5

» For the IDA* algorithm does it matter if the f function is non-monotonic?

> Yes

f function monotonicity – 6

» For the IDA* algorithm does it matter if the f function is non-monotonic?

> Yes

» Why?

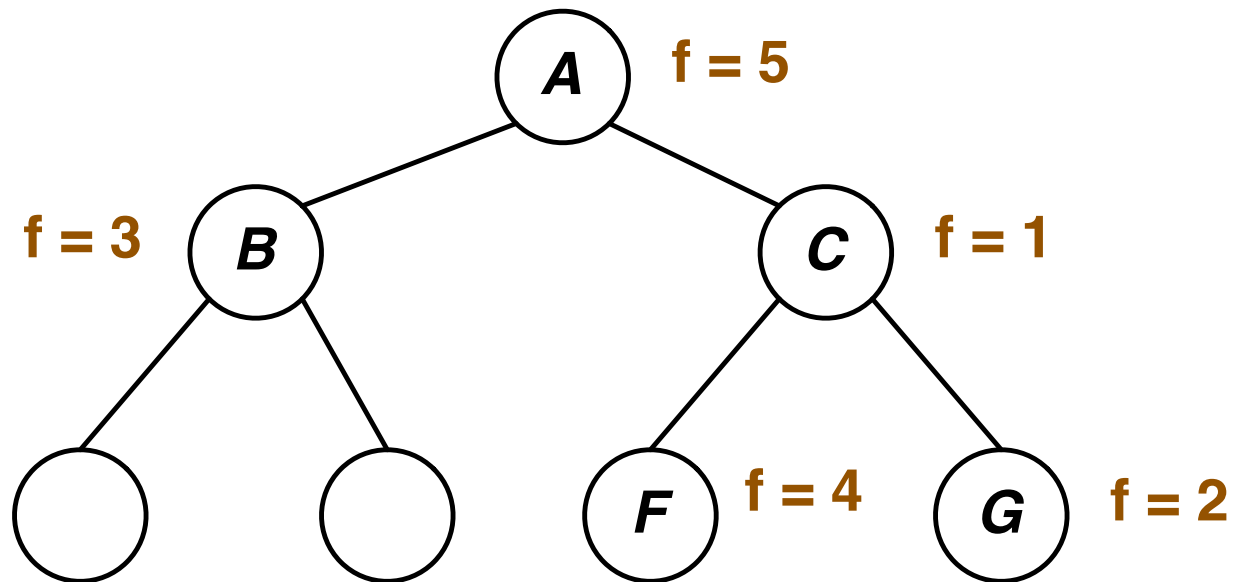
f function monotonicity – 7

- » **For the IDA* algorithm does it matter if the f function is non-monotonic?**
 - > **Yes**

- » **Why?**
 - > **The IDA* algorithm always expands paths from the start with a monotonically increasing f function it expands nodes in best-first order.**

Non-monotonic f function problem

- ◇ In the following if $f\text{-bound} = 3$, then the “B” node could expand before the “C”, “G” sequence



IDA* Problem

» **What is a major problem with IDA*?**

IDA* Problem – 2

- » **What is a major problem with IDA*?**
 - > **In unfavourable situations the cost of regenerating nodes becomes unacceptable**

IDA* Problem – 3

- » **What is a major problem with IDA*?**
 - > **In unfavourable situations the cost of regenerating nodes becomes unacceptable**

- » **How do we solve the problem?**

IDA* Problem – 4

- » **What is a major problem with IDA*?**
 - > **In unfavourable situations the cost of regenerating nodes becomes unacceptable**

- » **How do we solve the problem?**
 - > **Create a different algorithm**

IDA* Problem – 5

- » **What is a major problem with IDA*?**
 - > **In unfavourable situations the cost of regenerating nodes becomes unacceptable**

- » **How do we solve the problem?**
 - > **Create a different algorithm**
 - **RBFS – Recursive Best-First Search**