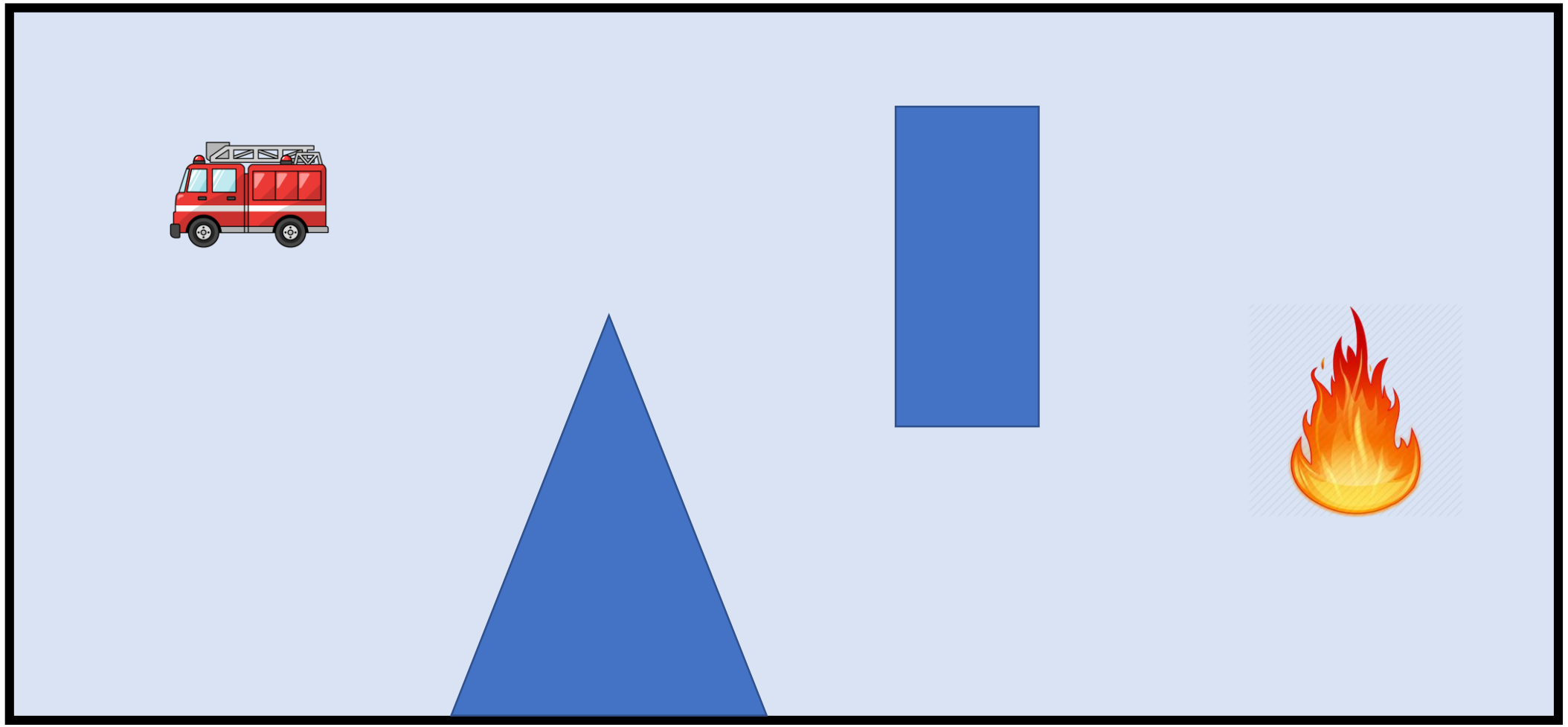
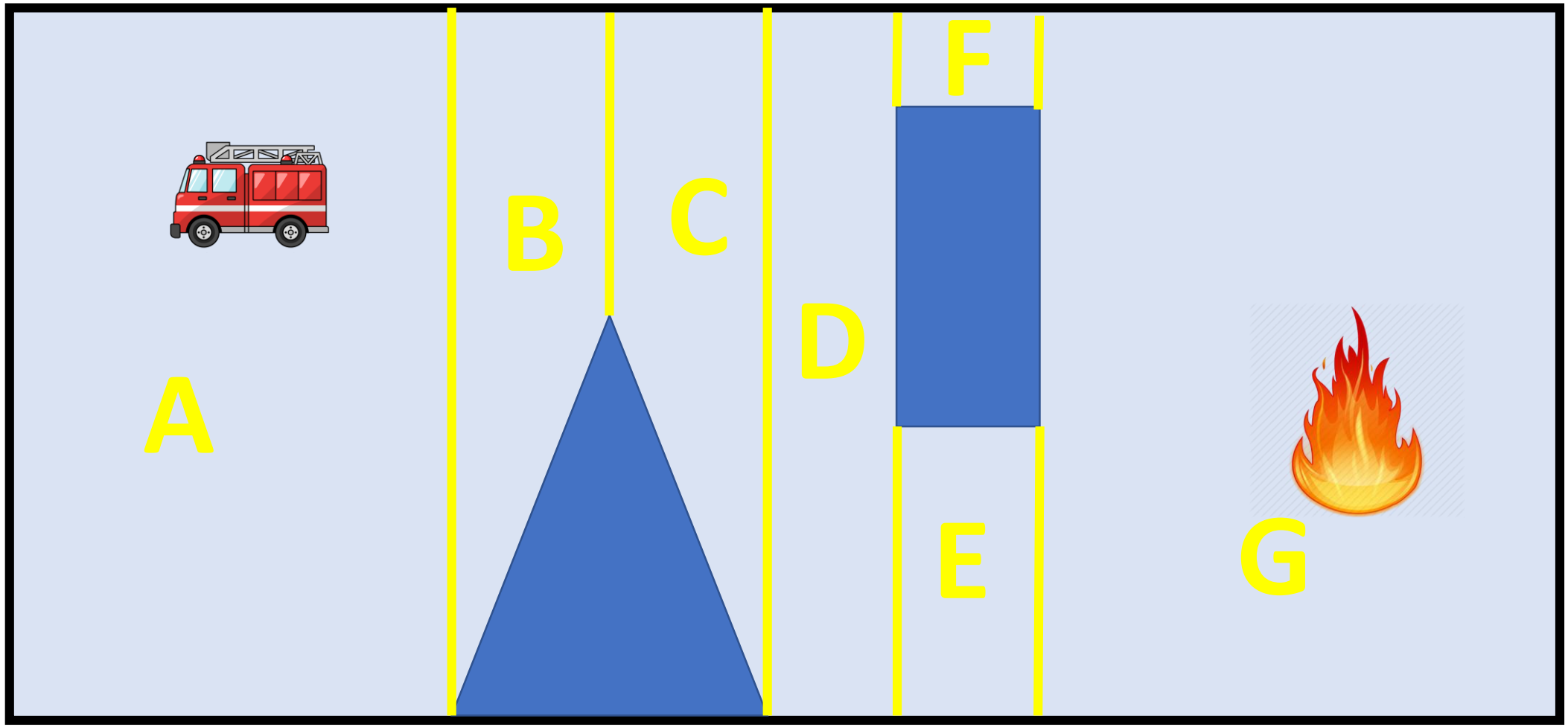
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

ΑΝΑΖΗΤΗΣΗ

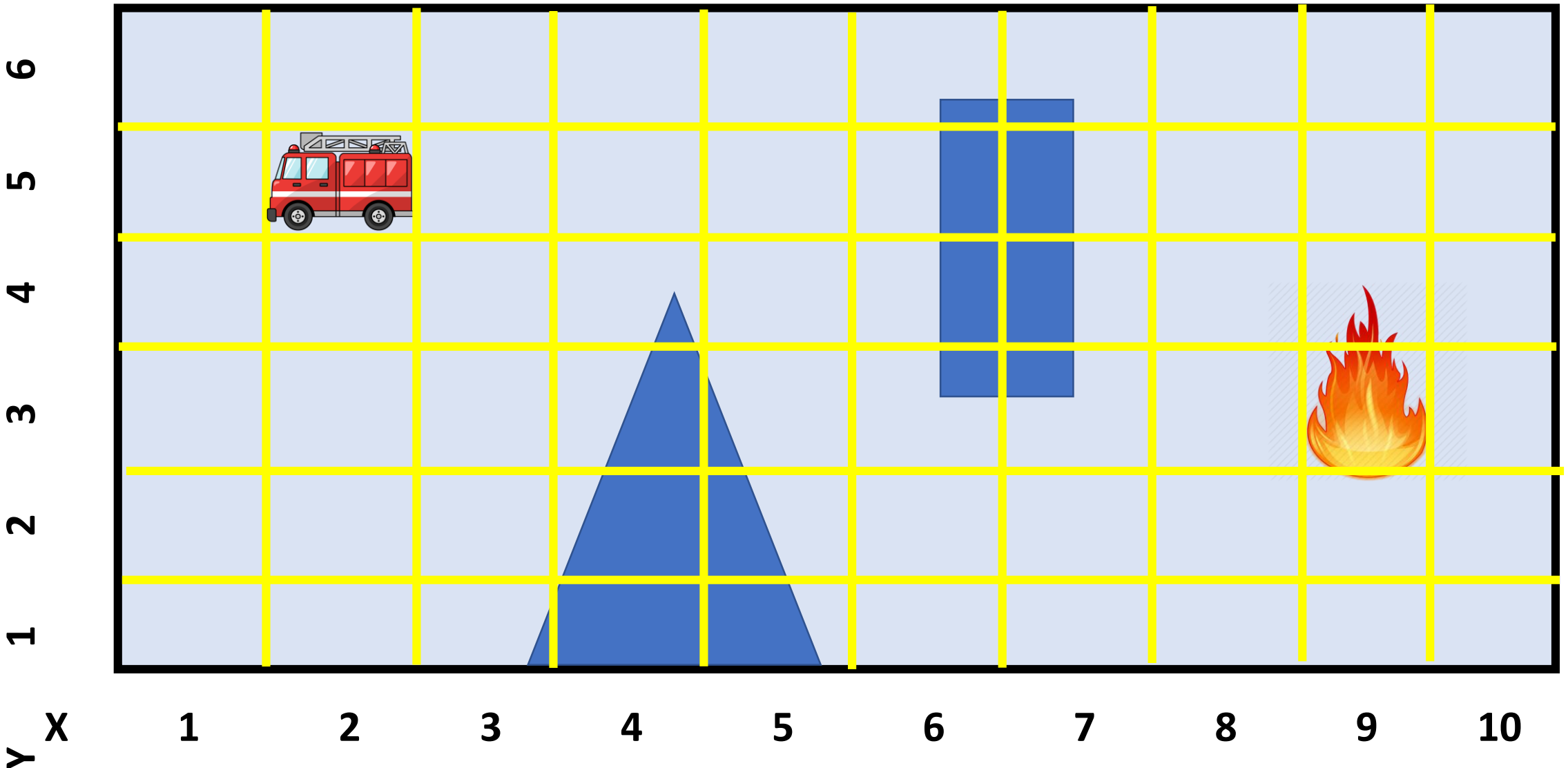
Πρόβλημα



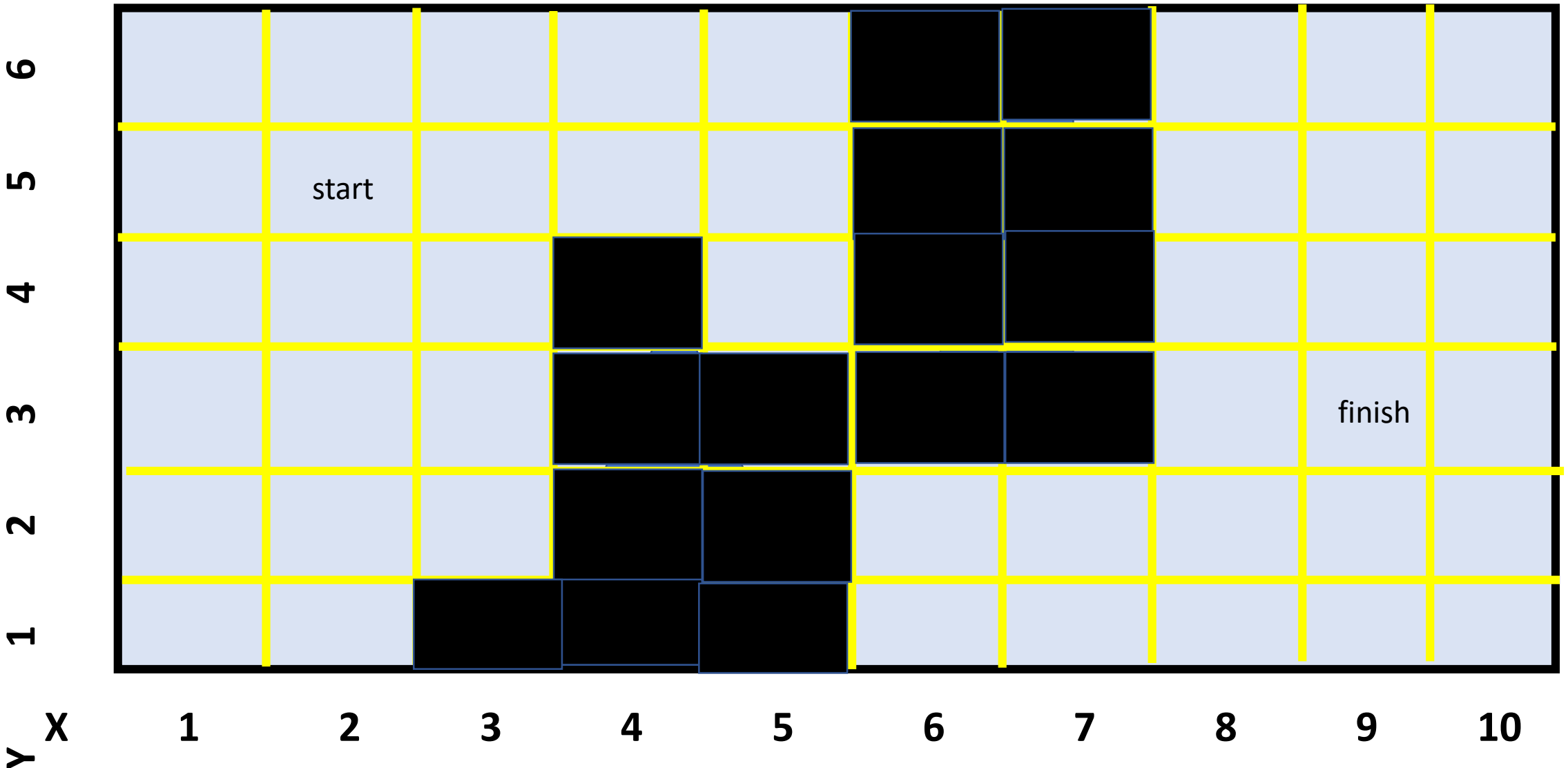
Exact Cell Decomposition



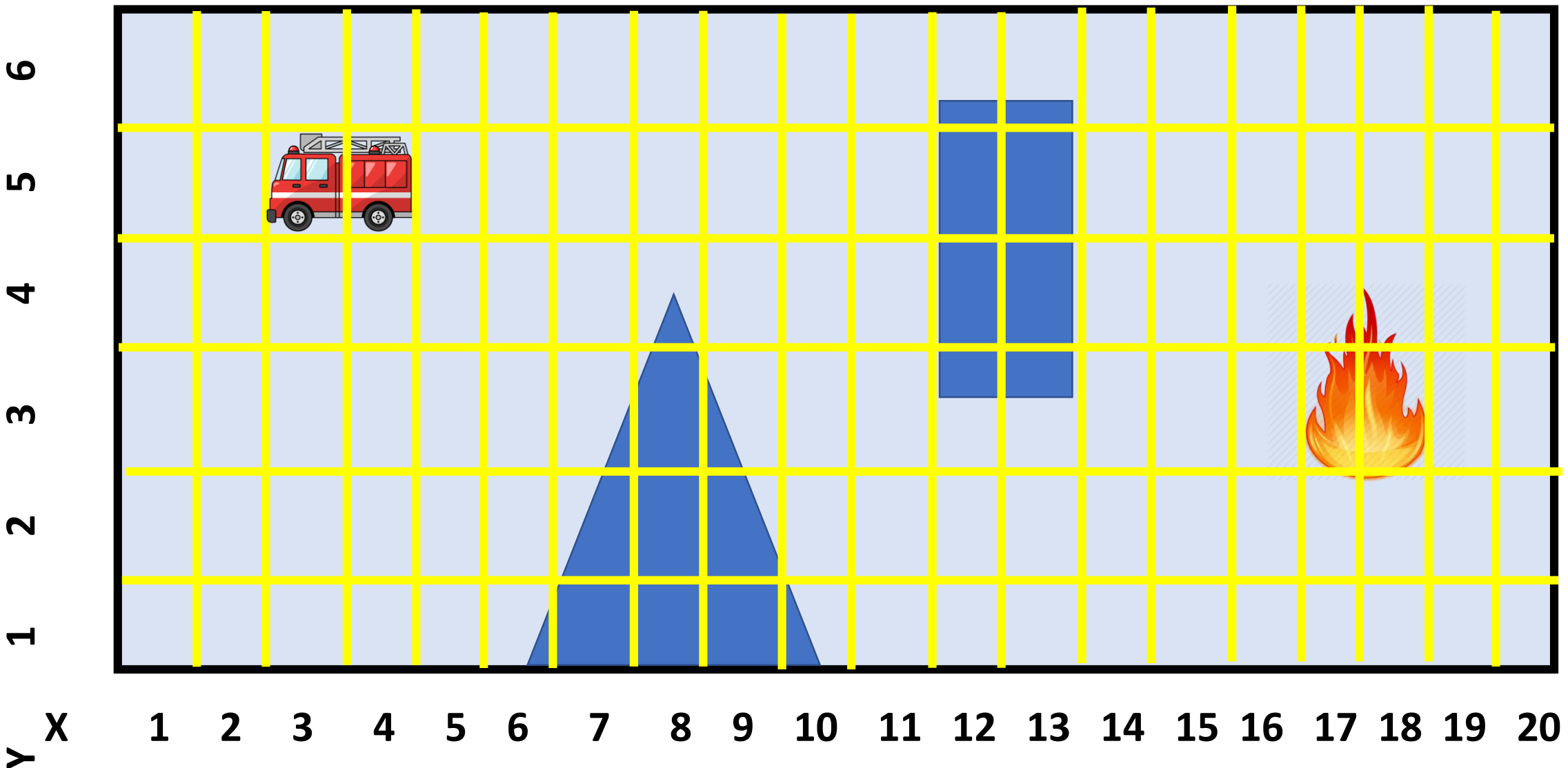
Fixed Cell Decomposition



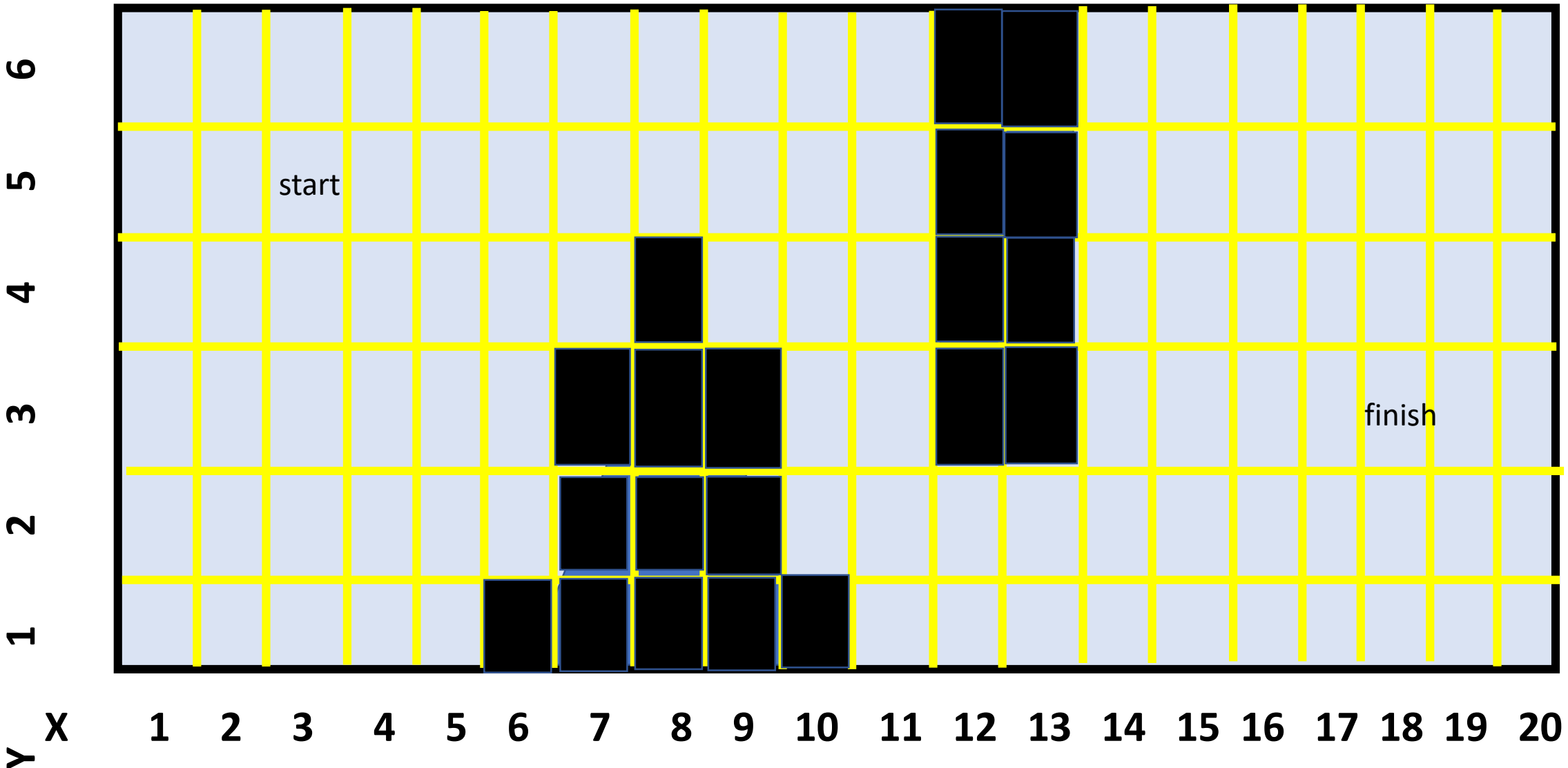
Unsolvable Problem



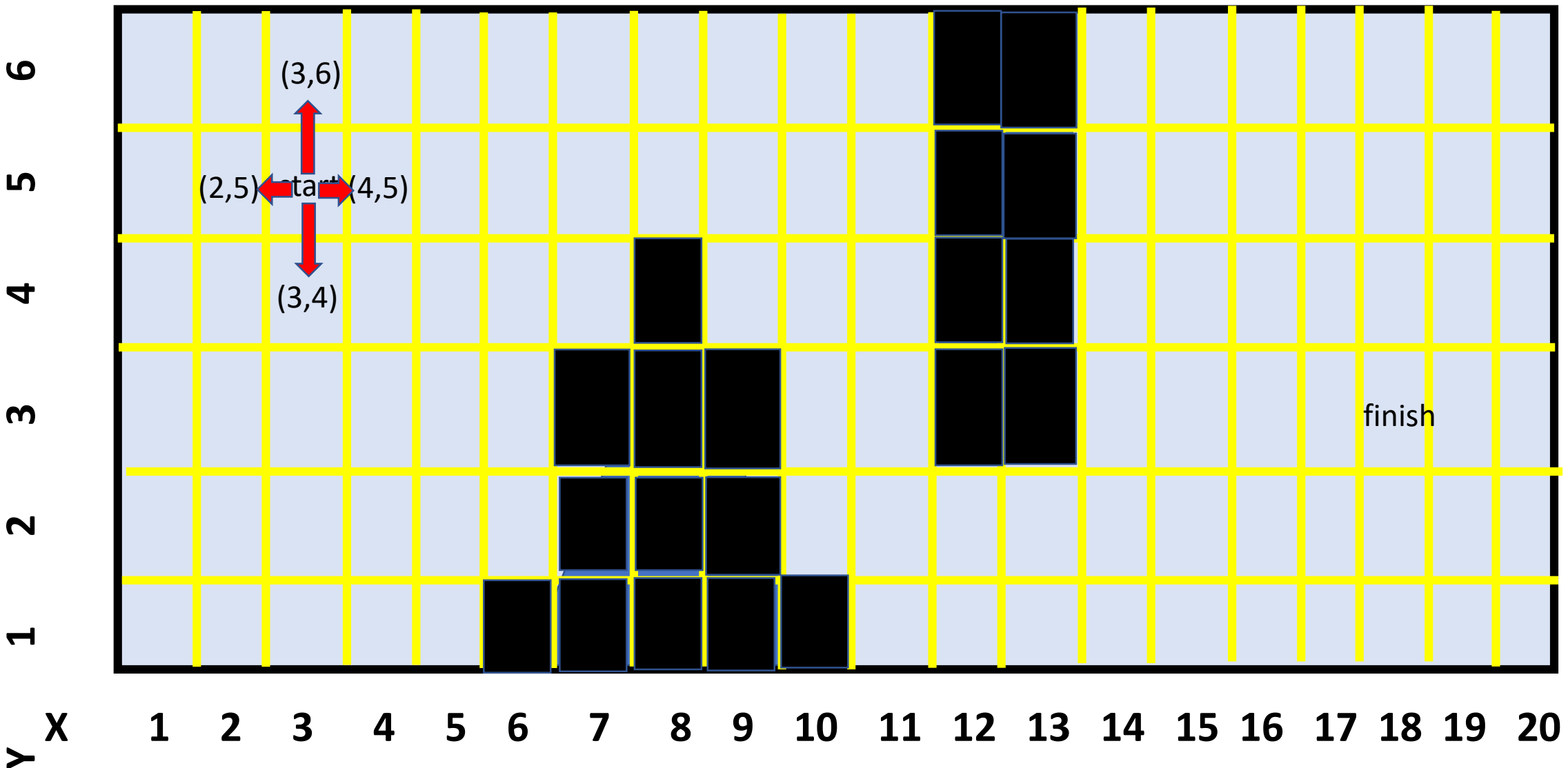
Smaller Cell



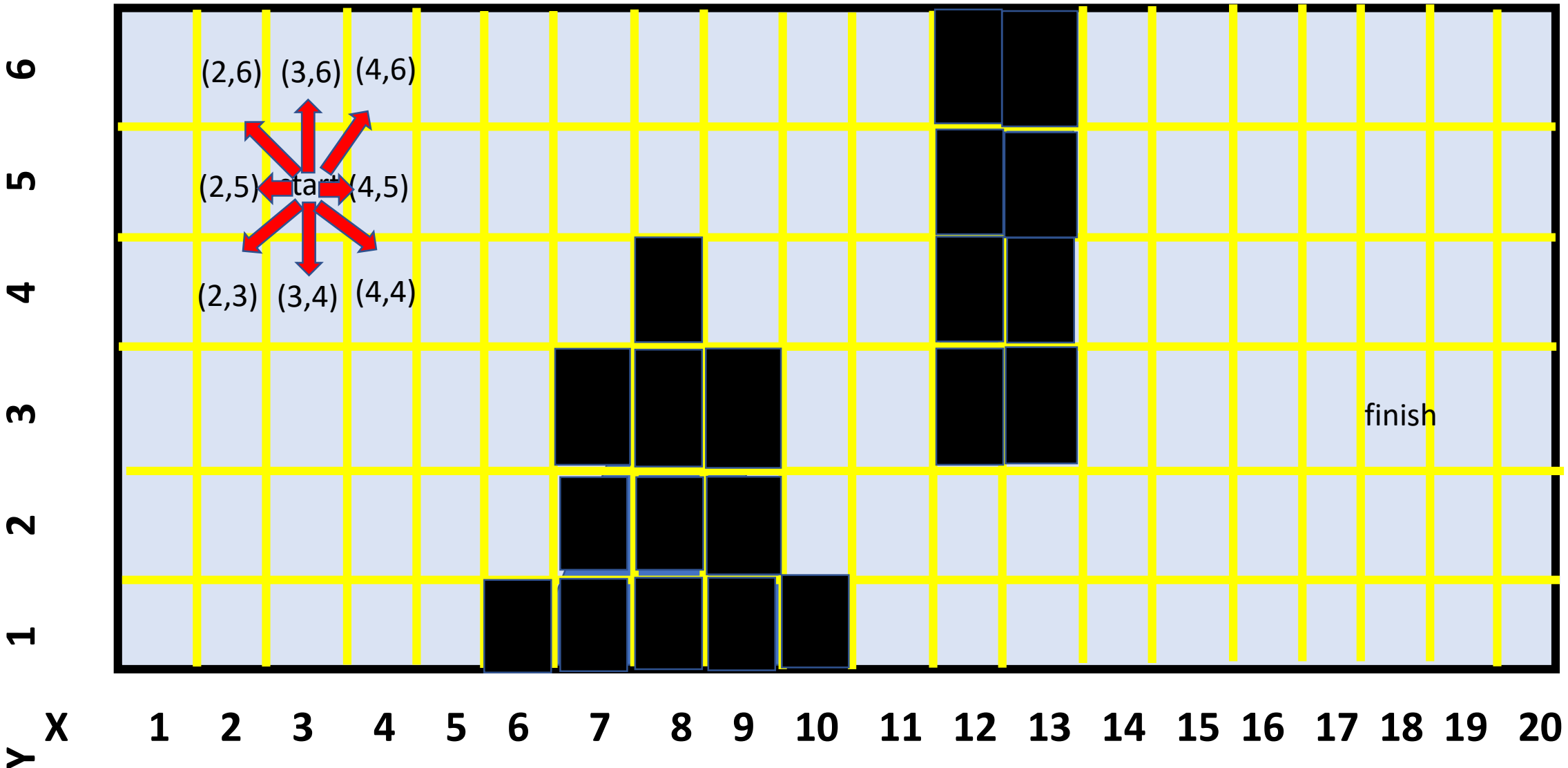
Solvable Problem



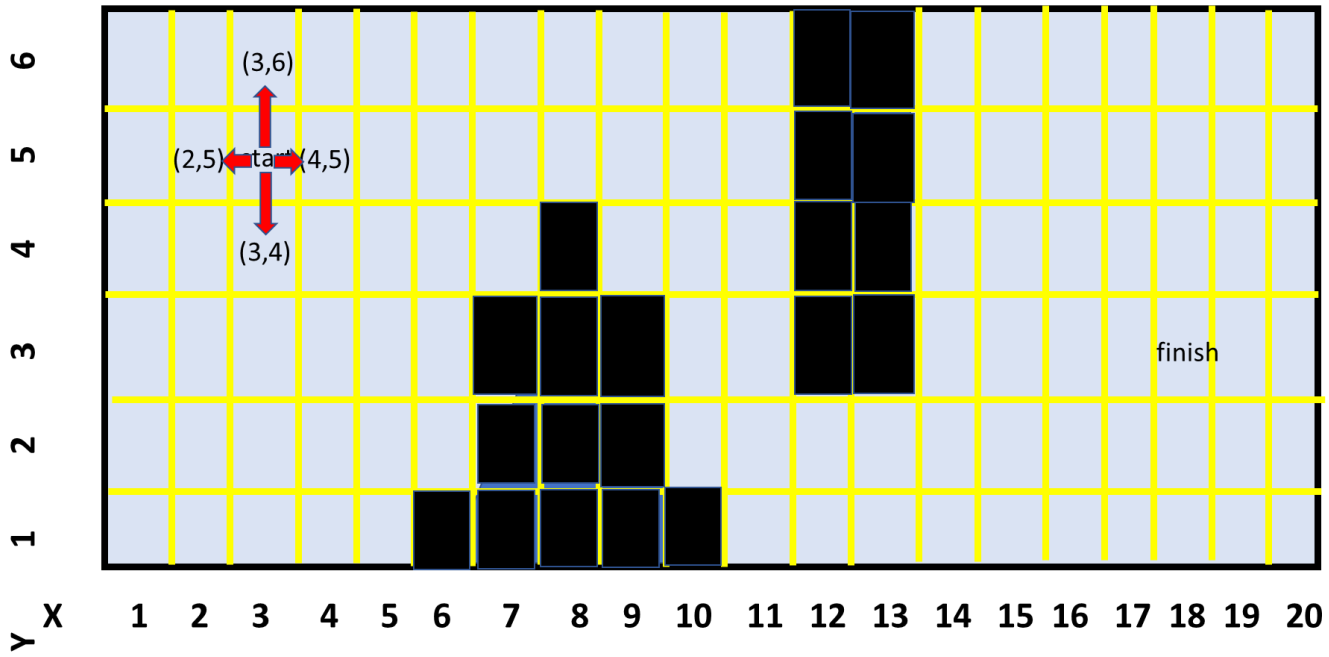
No diagonal movements

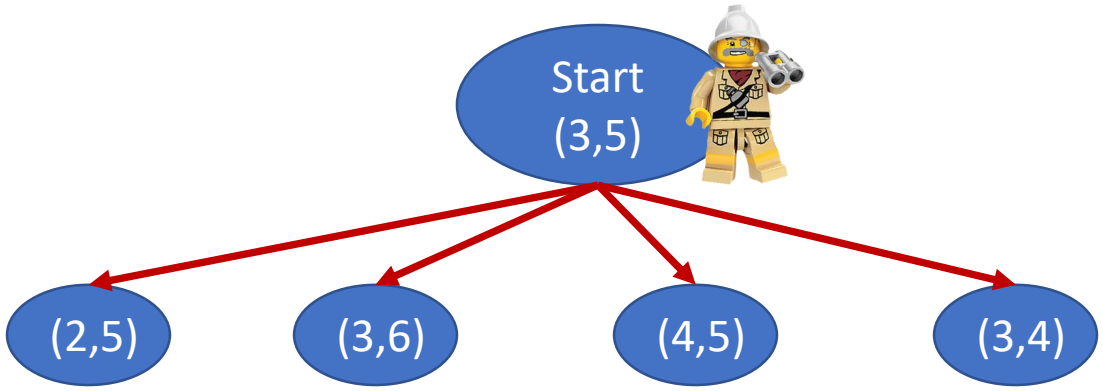


With diagonal movements

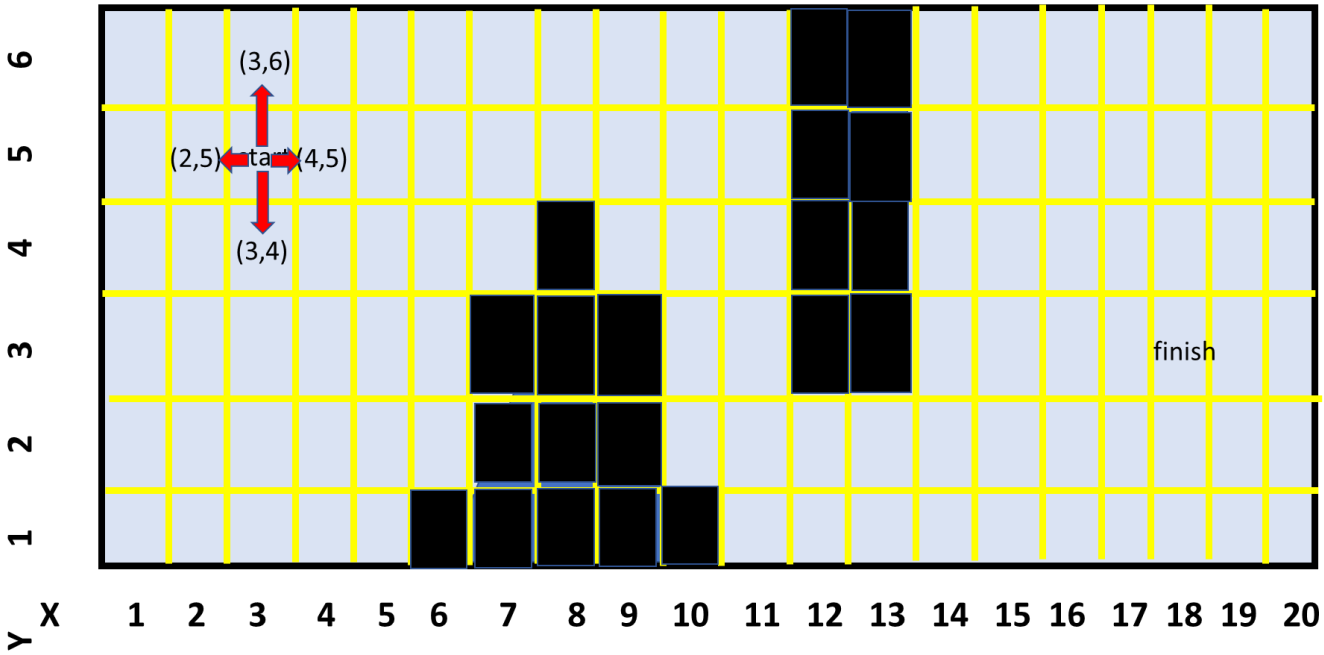


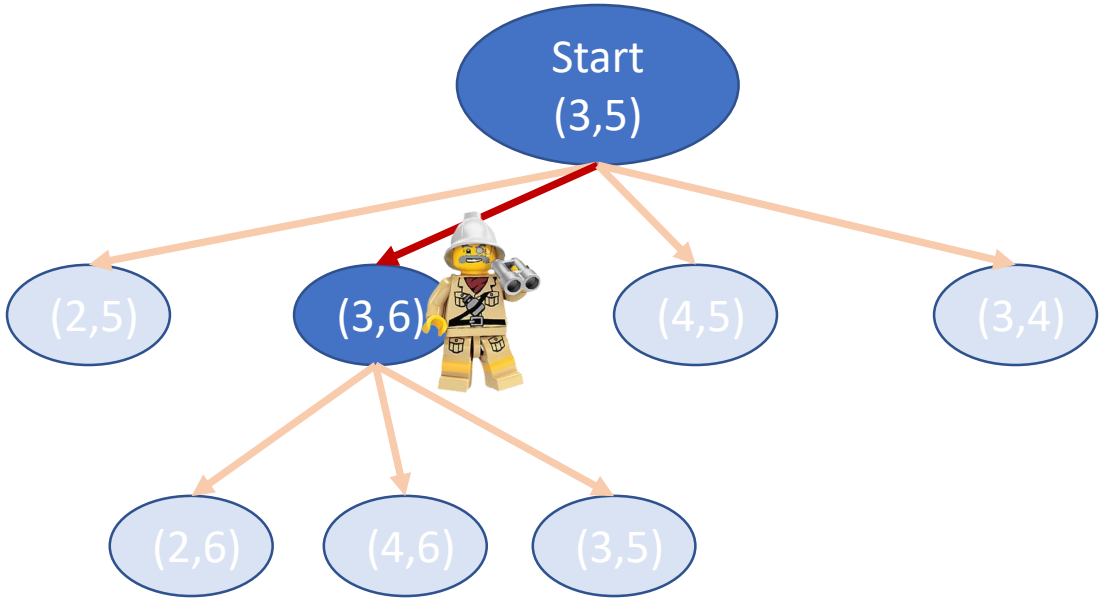
How do we traverse?



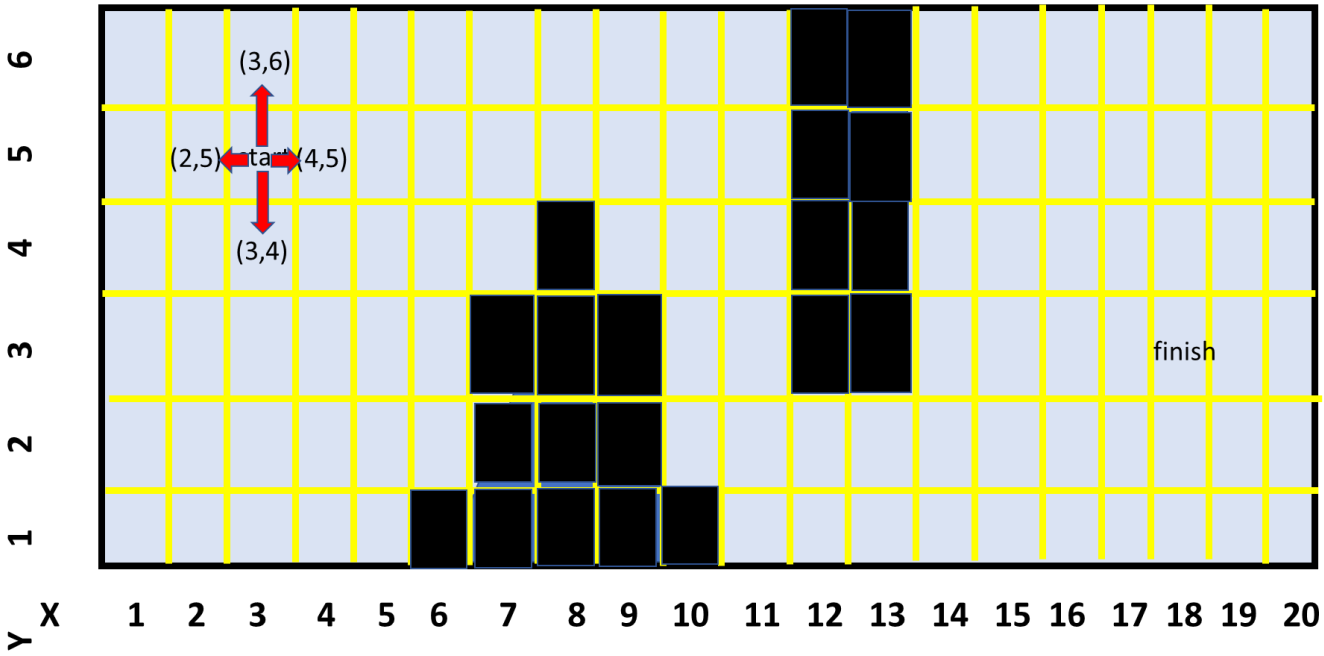


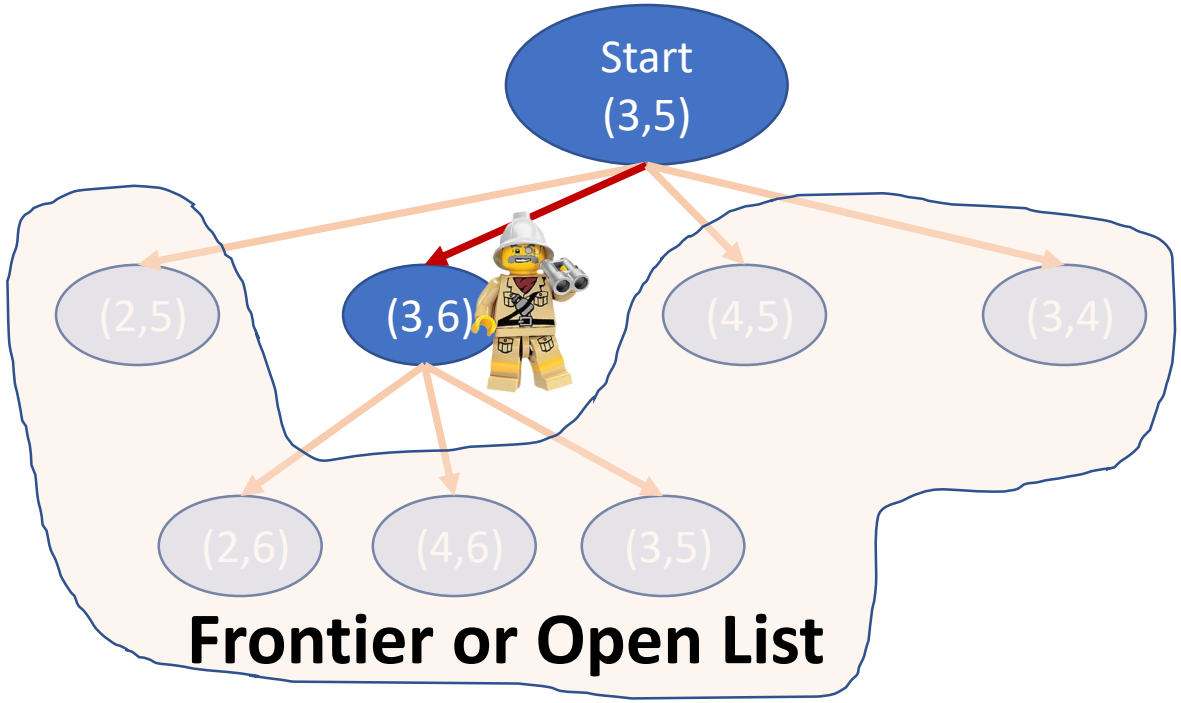
General Search Algorithm



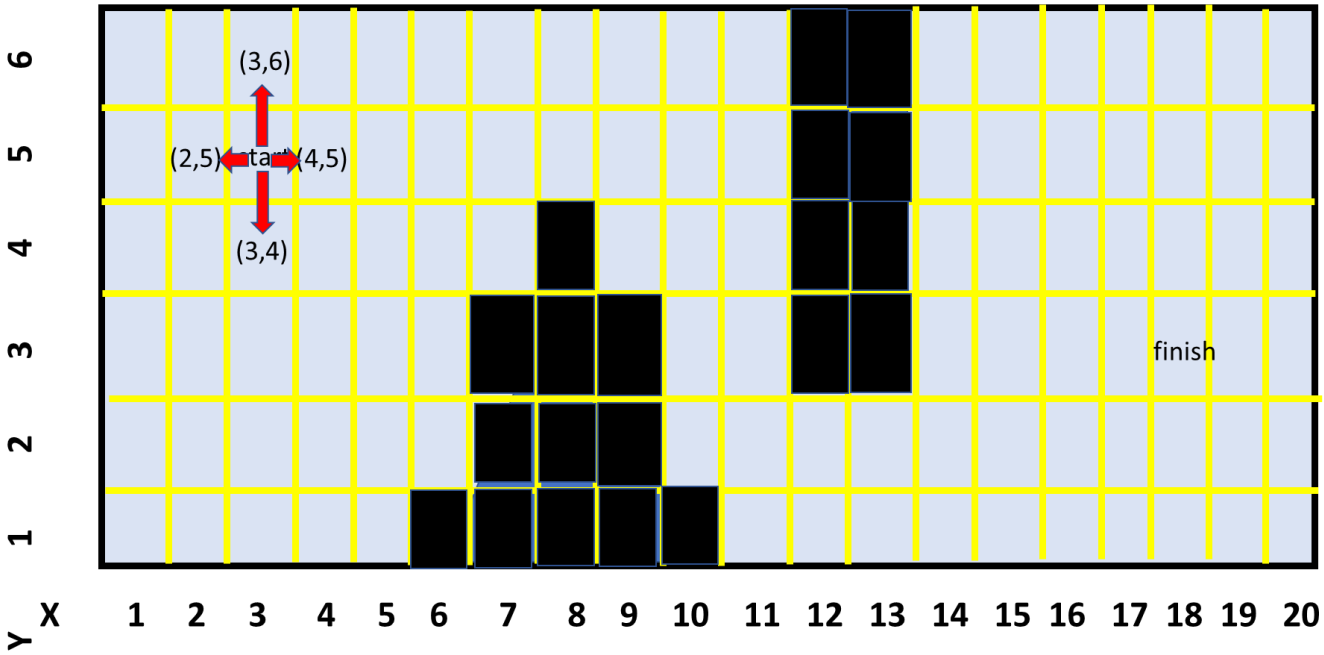


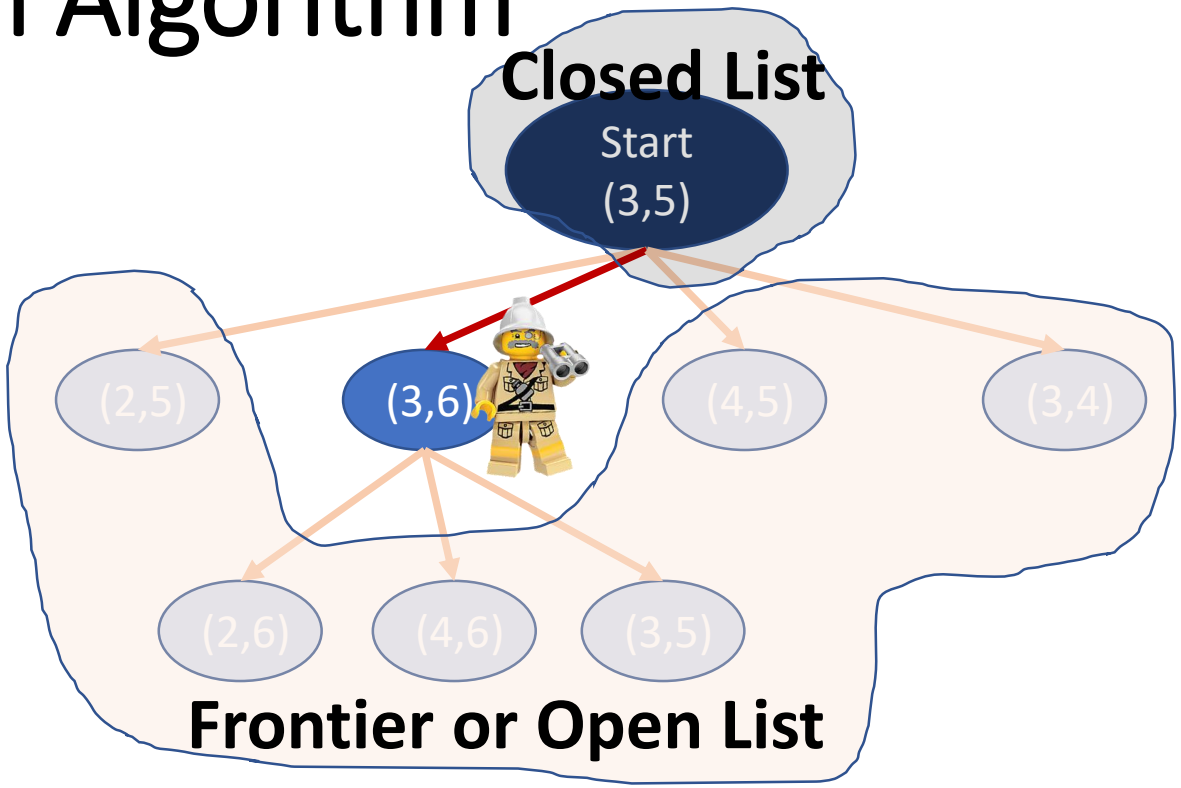
General Search Algorithm



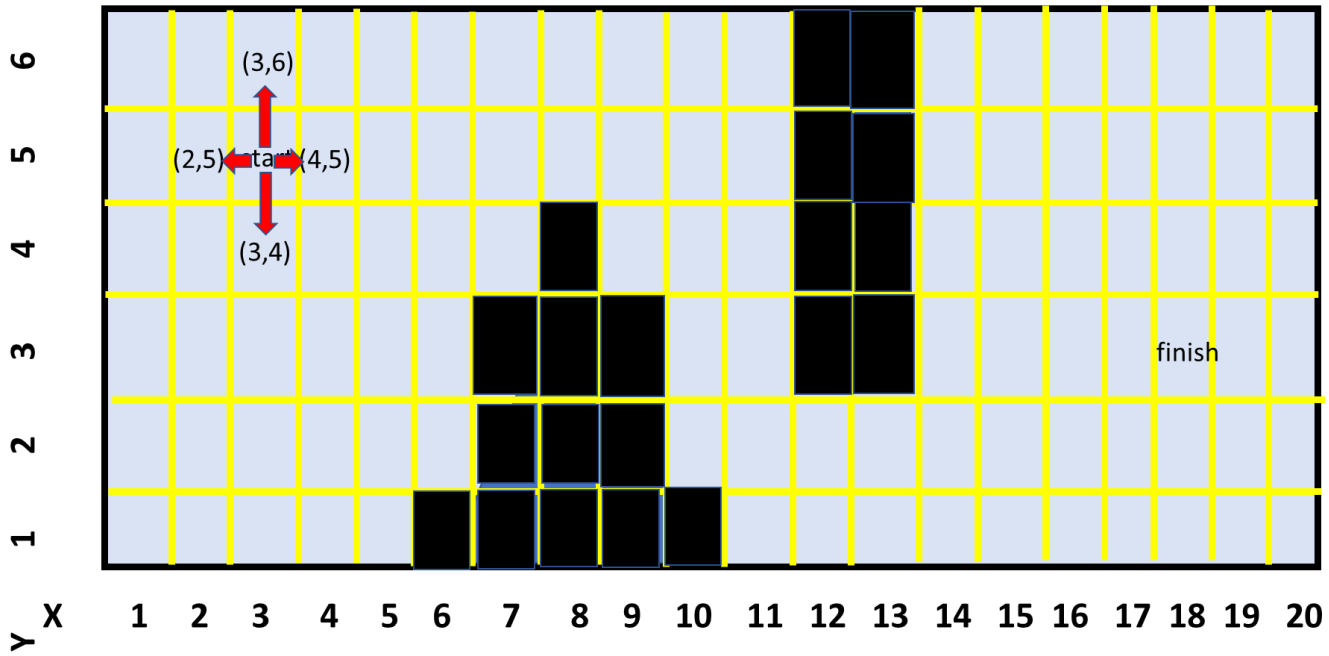


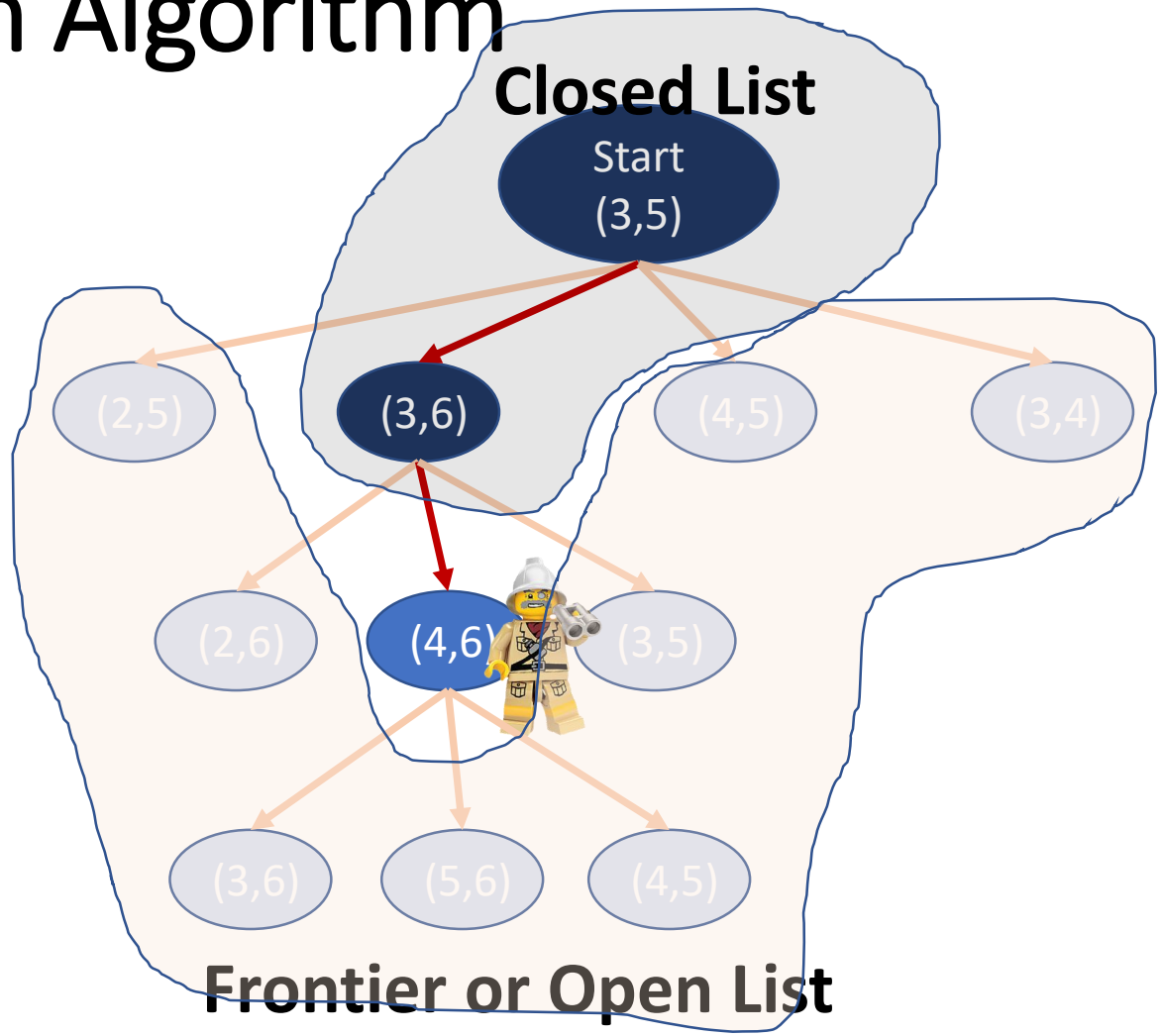
General Search Algorithm



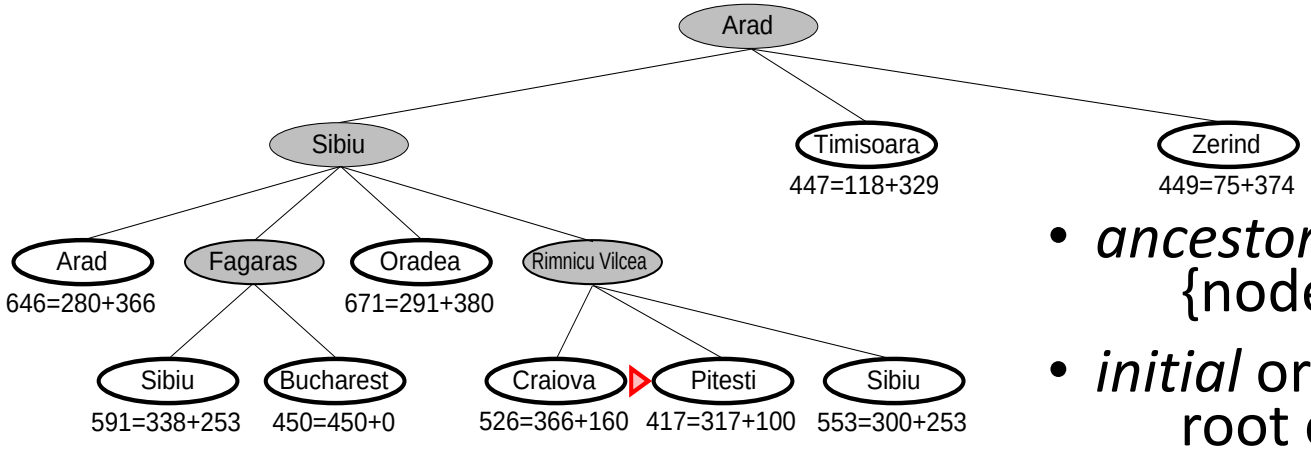


General Search Algorithm





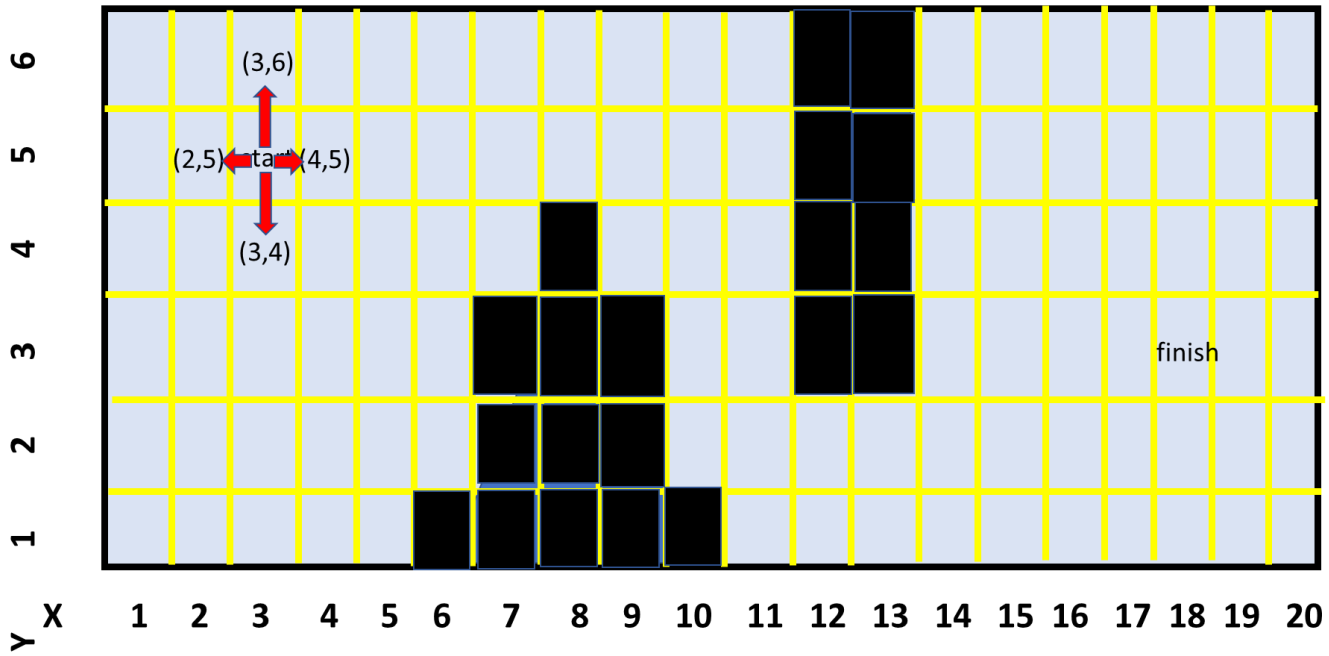
Search-Tree Terminology

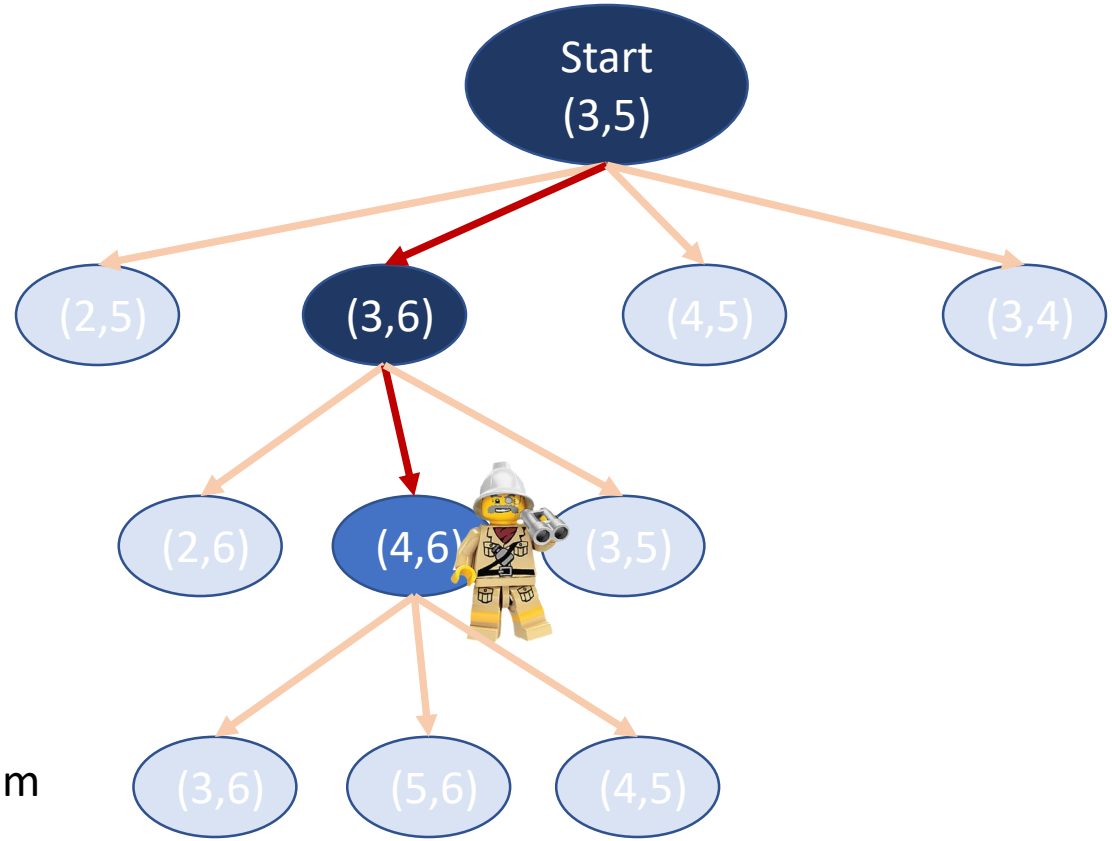


- *Node*: a pair ν = (π,s), where s = γ(s_0,π)
- In practice, ν may contain other things
  - pointer to parent, cost(π), …
  - π not always stored explicitly, can be computed from the parent pointers
- *children* of ν = {($\pi.a$, $\gamma(s,a)$) | a is applicable in s}
- *successors* of ν: children, children of children, etc.

- *ancestors* of ν = {nodes that have ν as a successor}
- *initial* or *starting* node: $\nu_0 = (\langle\rangle, s_0)$ root of the search tree
- *path* in the search space: sequence $\langle \nu_0, \nu_1, \ldots, \nu_n \rangle$ such that each ν_i is a child of ν_{i-1}
- *height* of search space = length of longest acyclic path from ν_0
- *depth* of ν = length(π) = length of path from ν_0 to ν
- *branching factor* of ν = number of children
- *branching factor* of search tree = max branching factor of the nodes
- *expand* ν: generate all children

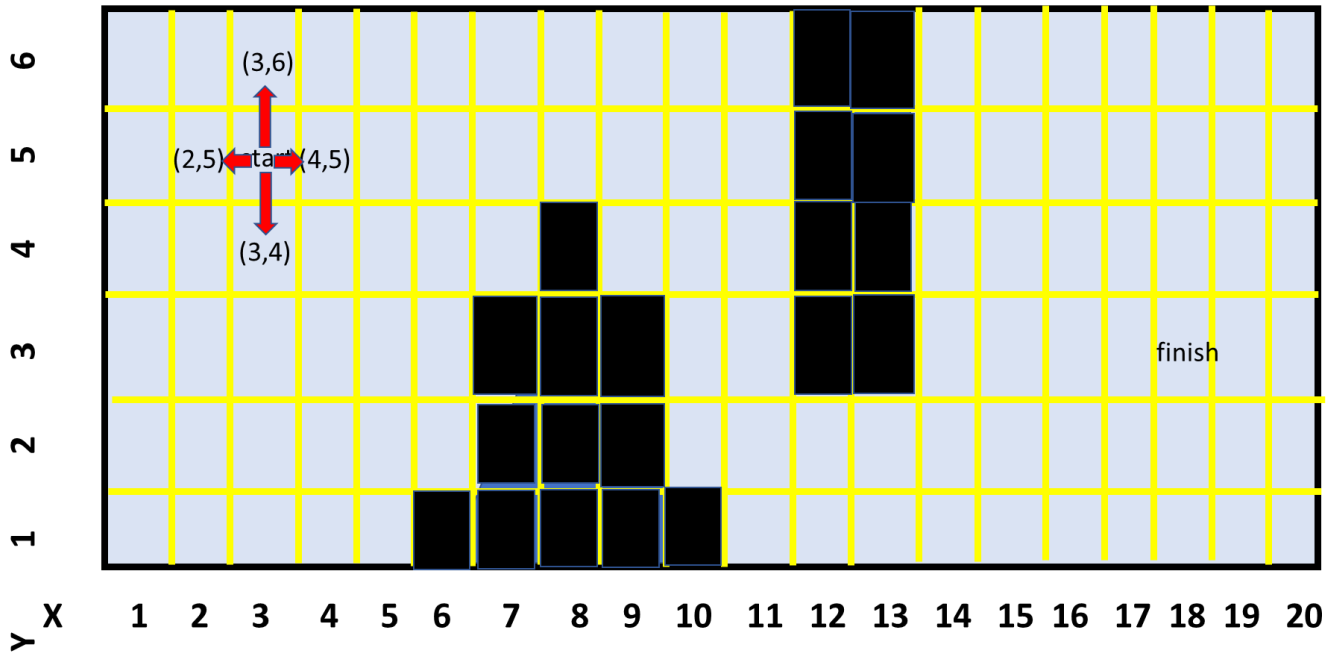
Criteria of Search Algorithms

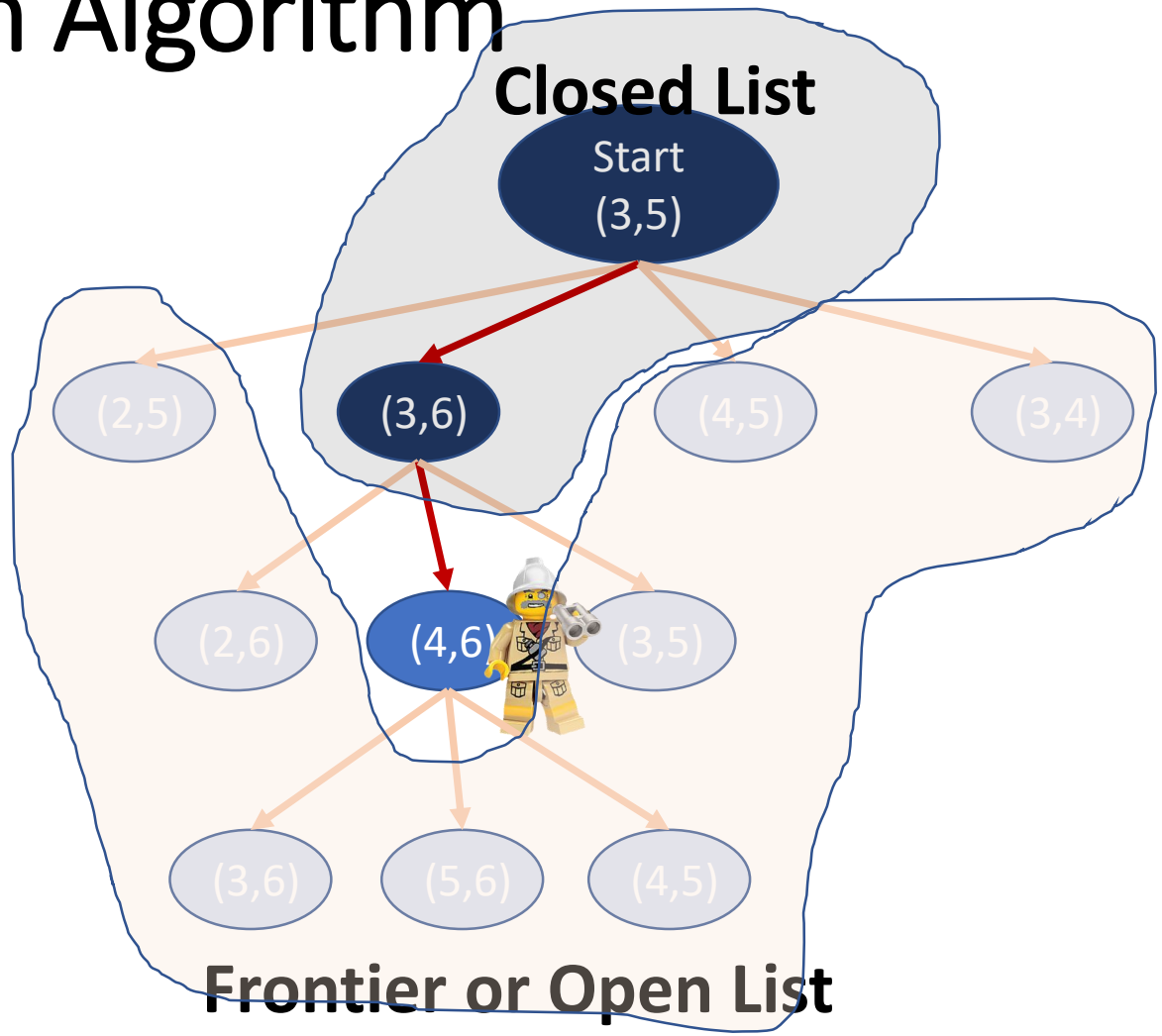




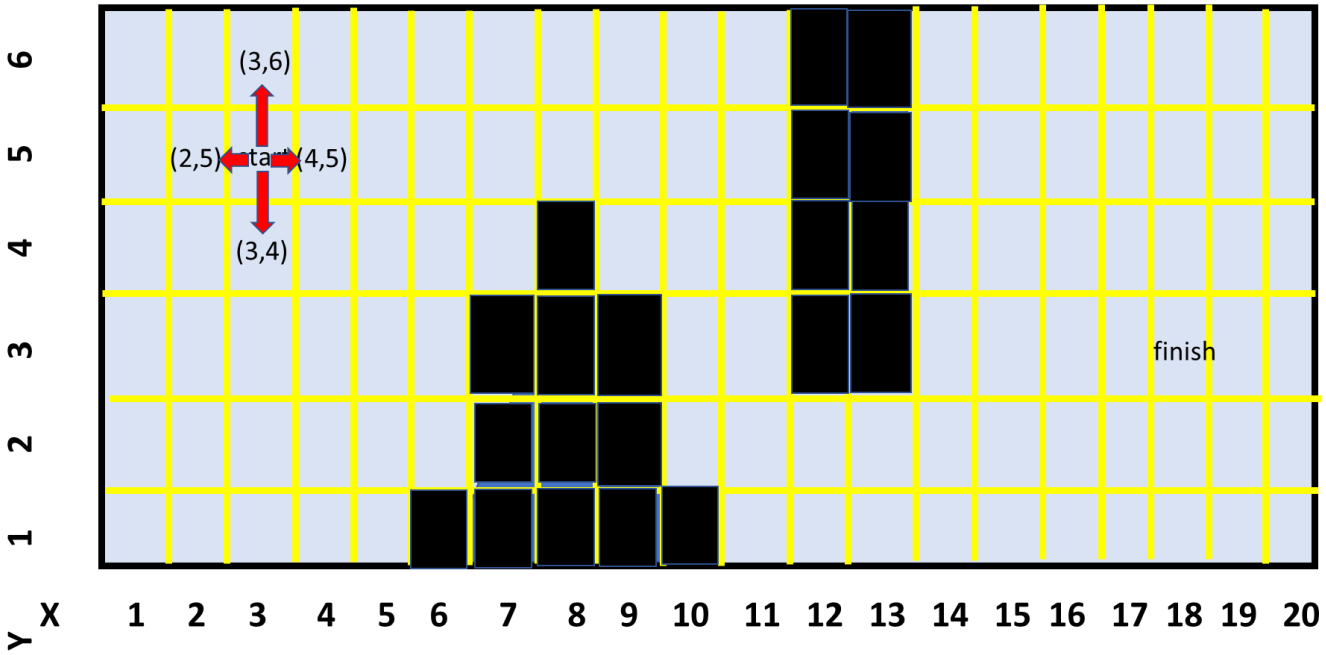
- **Soundness** (Ορθότητα) : All the solutions returned by the algorithm are feasible
- **Completeness** (Πληρότητα): If the problem is solvable the algorithm will return at least one solution
- **Optimality** ("Βελτιστότητα"???): The first solution returned by the algorithm is the optimal ← Optimality Criterion

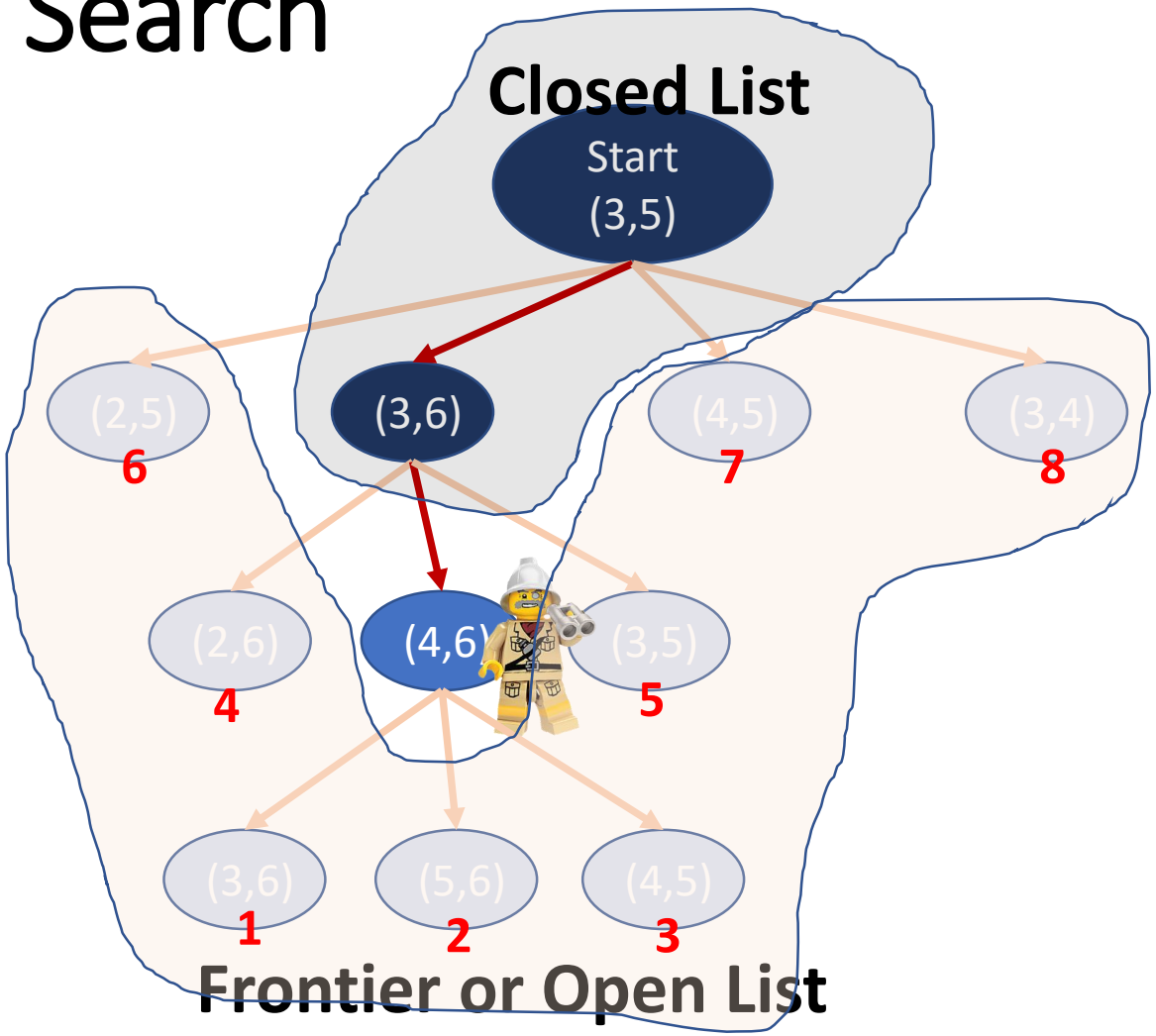
General Search Algorithm





Depth First Search



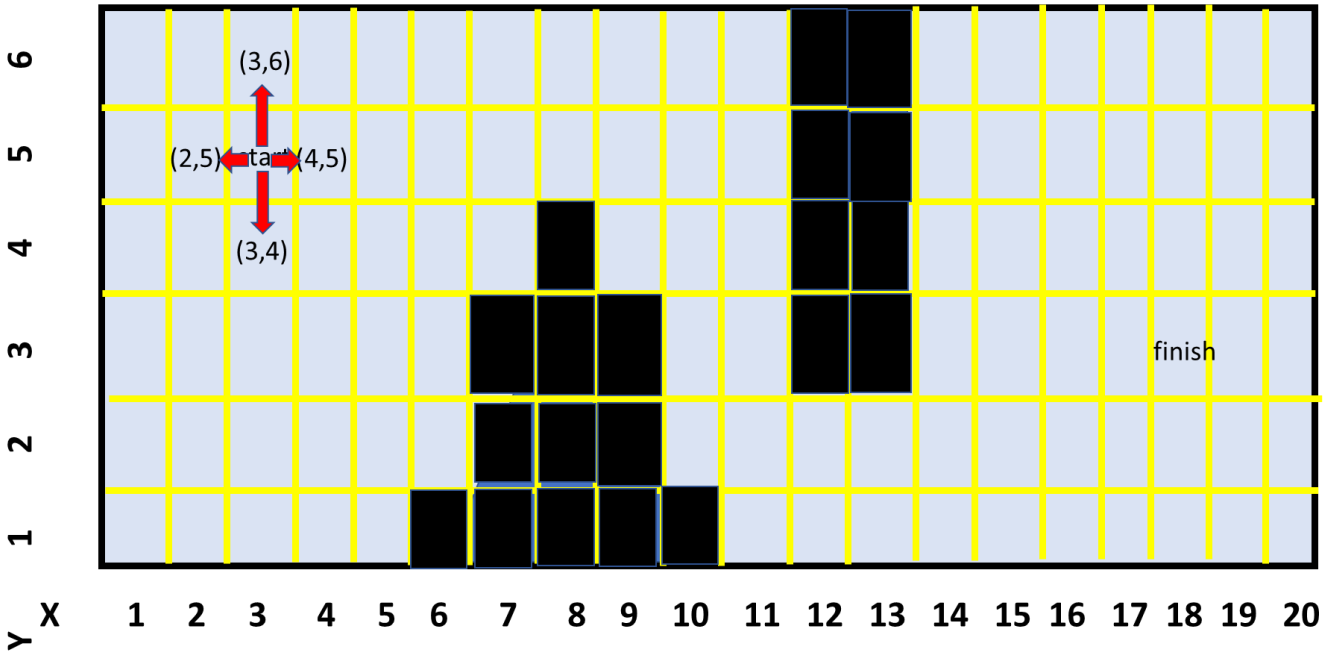


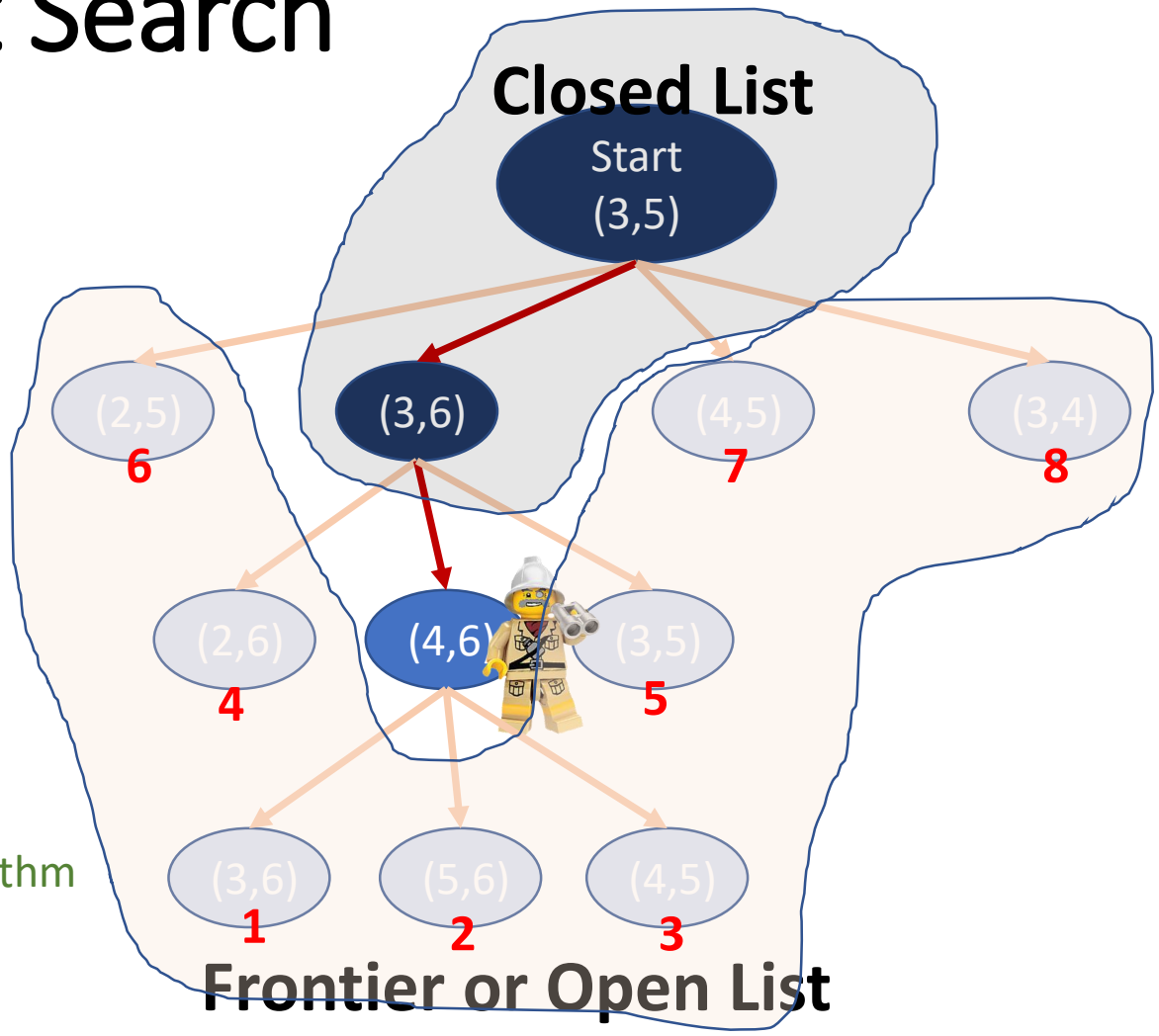
New States at the Beginning of the Open List

Ο Αλγόριθμος DFS

- Βάλε την **αρχική κατάσταση** στο **μέτωπο της αναζήτησης**.
- Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
- Βγάλε την πρώτη **κατάσταση** από το μέτωπο της αναζήτησης.
- Αν η κατάσταση ανήκει στο **κλειστό σύνολο** τότε πήγαινε στο βήμα 2.
- Αν η κατάσταση είναι μία από τις **τελικές**, τότε ανέφερε τη **λύση**.
- Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
- Εφάρμοσε τους **τελεστές μετάβασης** για να βρεις τις **καταστάσεις-παιδιά**.
- Βάλε τις καταστάσεις-παιδιά στην αρχή του μετώπου της αναζήτησης.
- Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
- Πήγαινε στο βήμα 2.

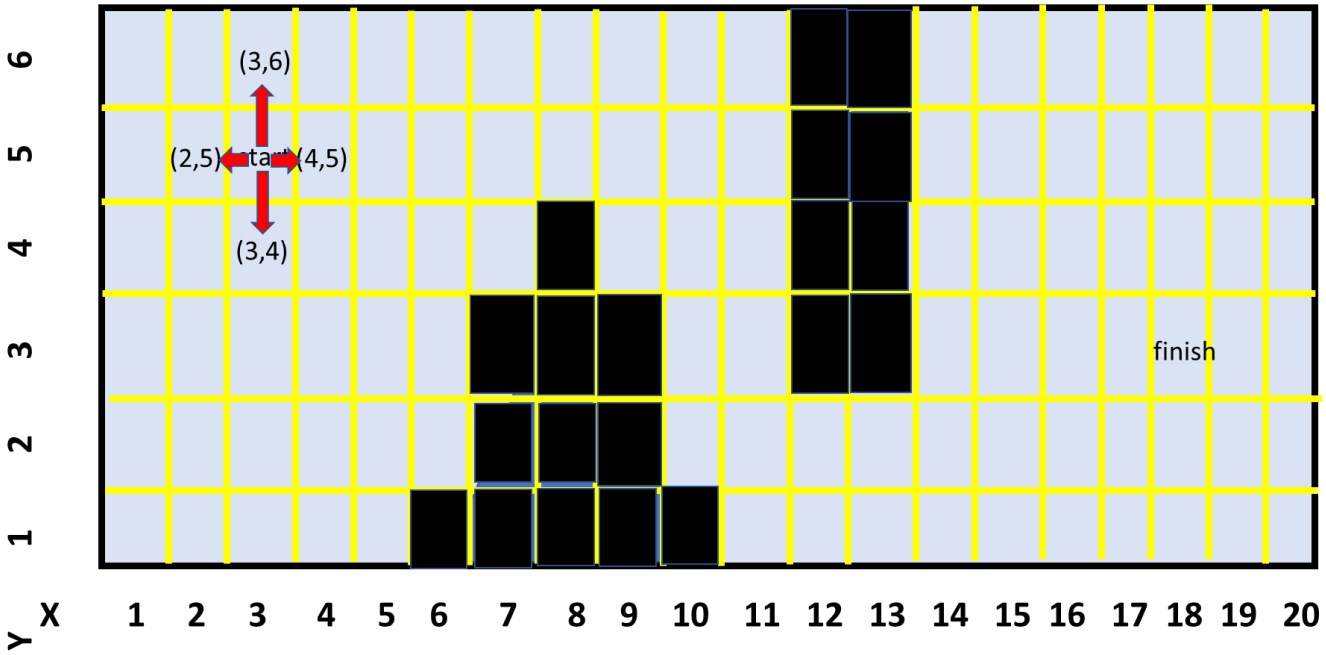
Depth First Search

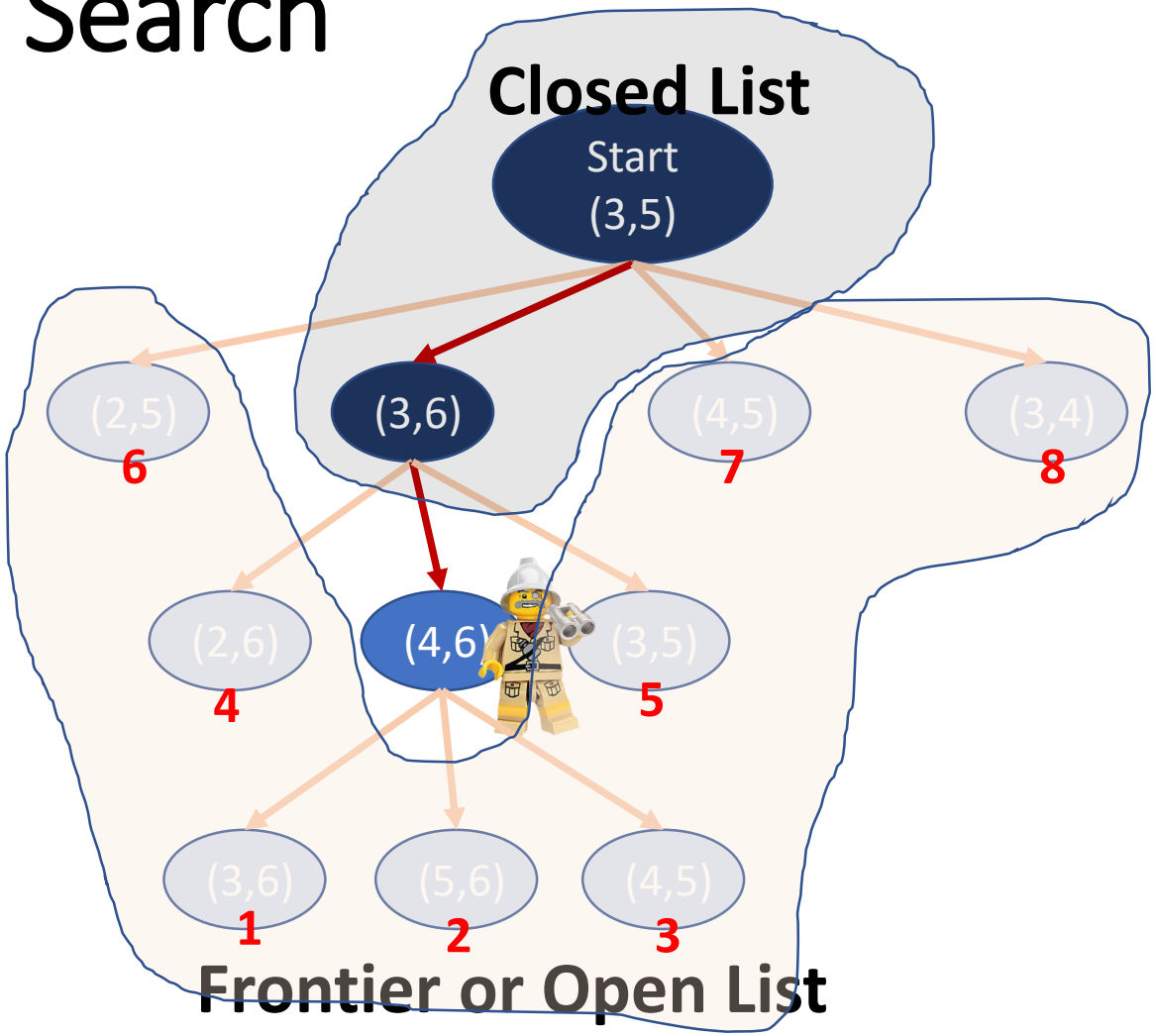




- **Soundness** (Ορθότητα) : All the solutions returned by the algorithm are feasible
- **Completeness** (Πληρότητα): If the problem is solvable the algorithm will return at least one solution
- **Optimality** ("Βελτιστότητα"???): The first solution returned by the algorithm is the optimal ← Optimality Criterion

Depth First Search





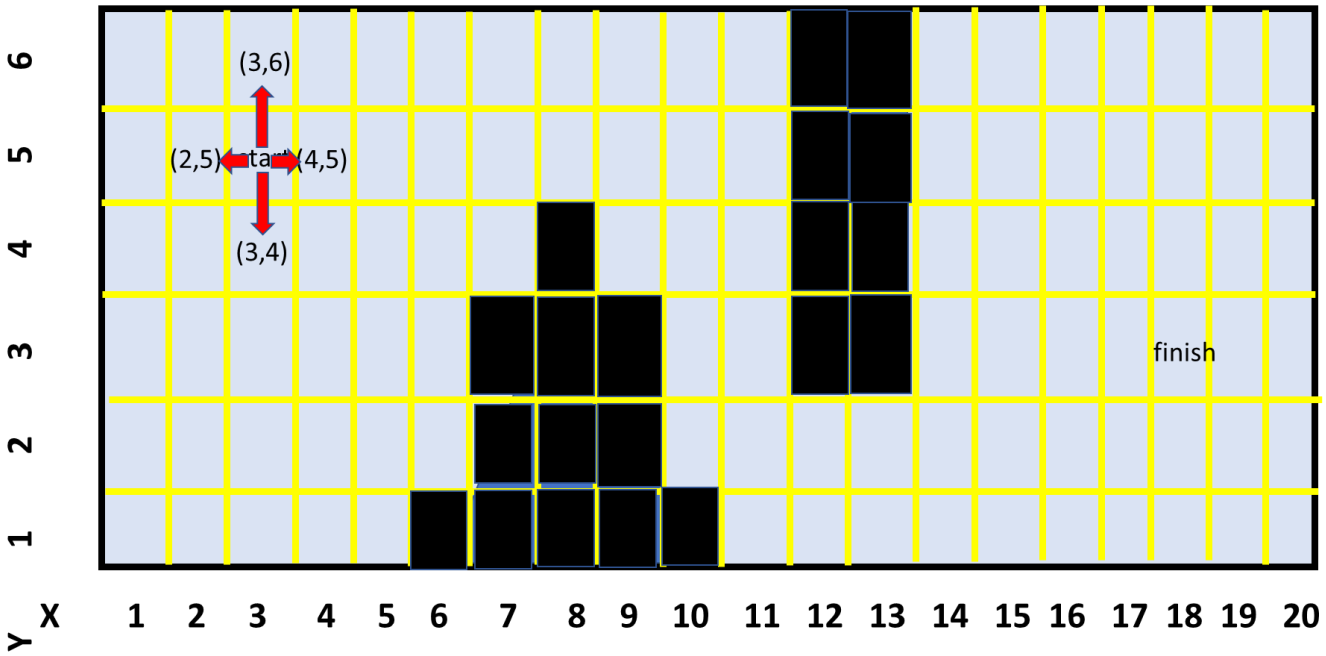
Worst-case running time $O(b^l)$

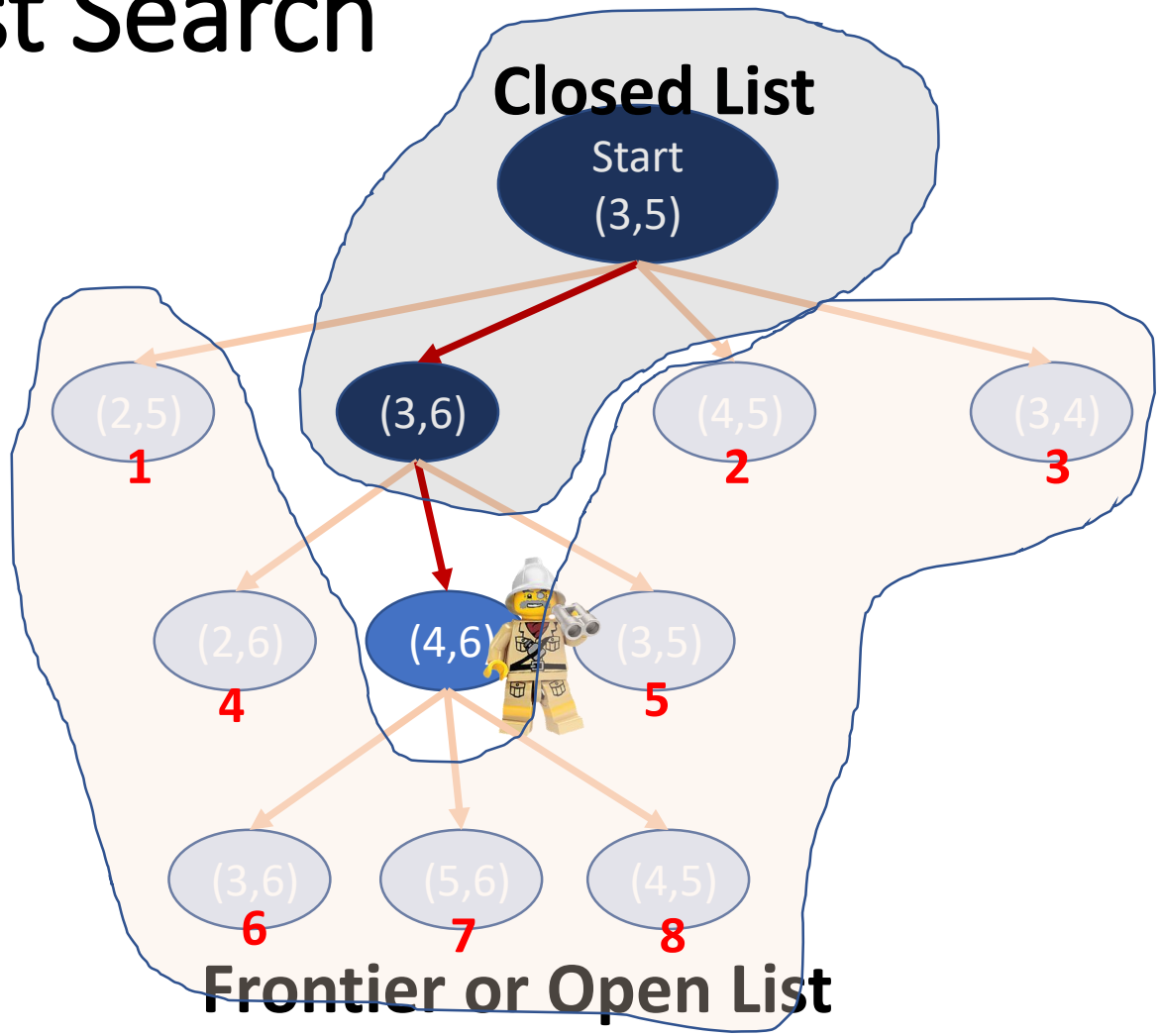
Worst-case memory $O(bl)$

b = max branching factor

l = max depth of any node

Breadth First Search



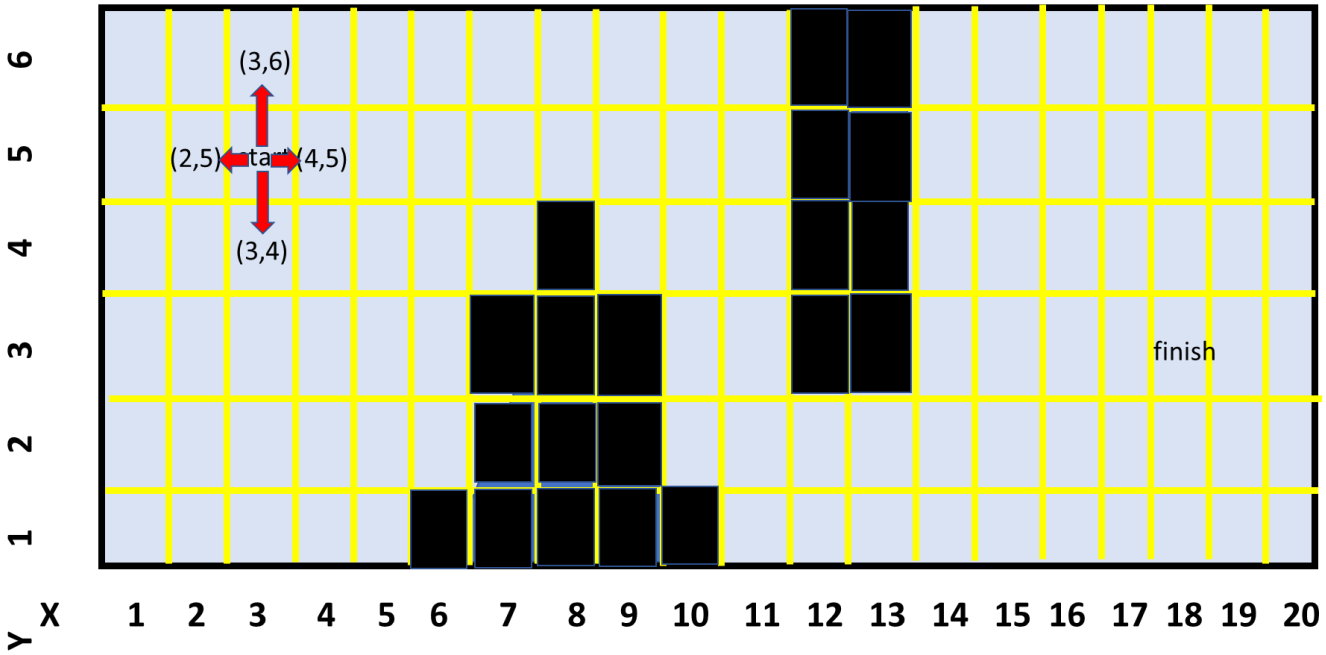


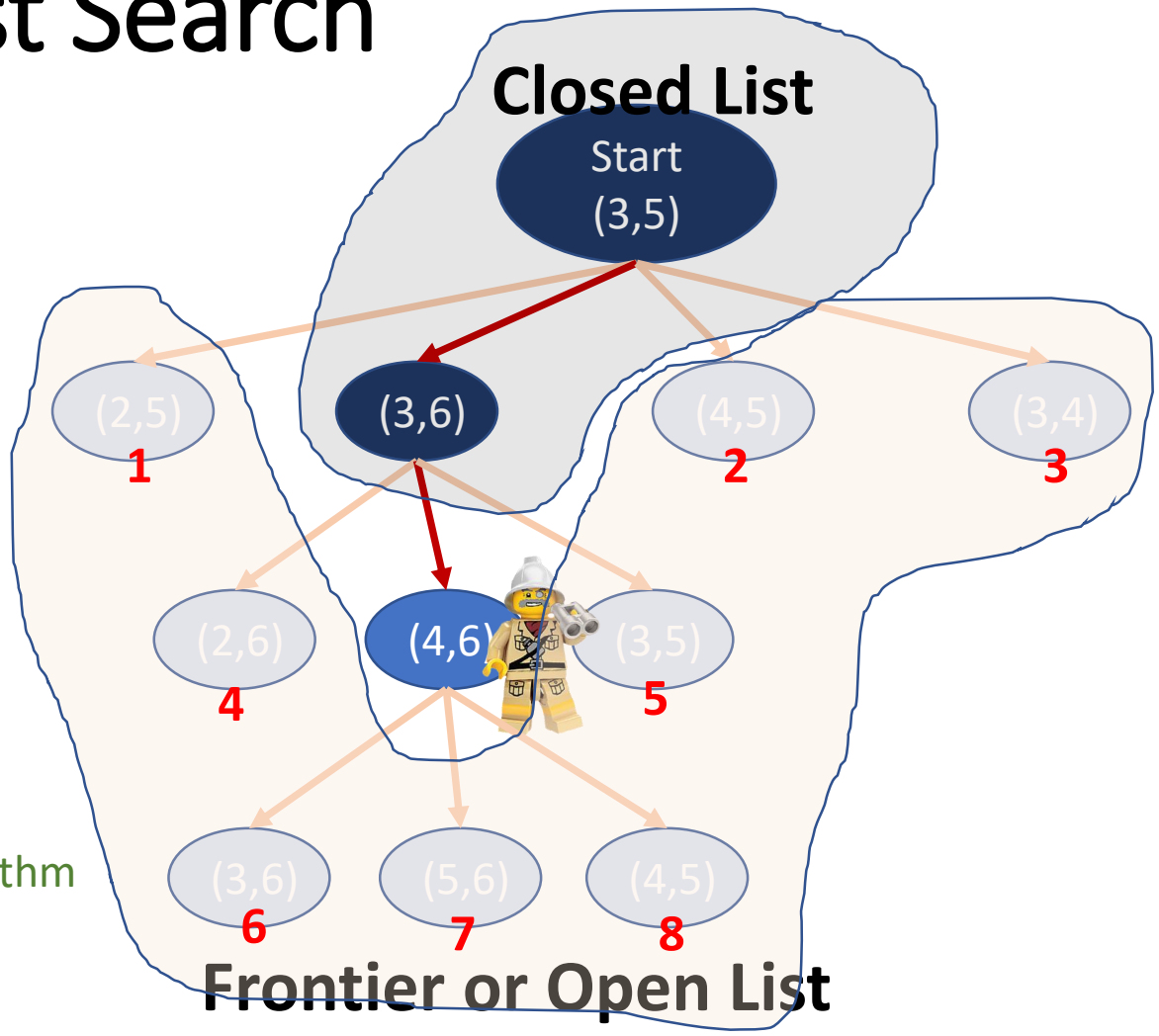
New States at the End of the Open List

Ο Αλγόριθμος BFS

- Βάλε την **αρχική κατάσταση** στο **μέτωπο της αναζήτησης**.
- Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
- Βγάλε την πρώτη **κατάσταση** από το μέτωπο της αναζήτησης.
- Αν η κατάσταση ανήκει στο **κλειστό σύνολο** τότε πήγαινε στο βήμα 2.
- Αν η κατάσταση είναι μία από τις **τελικές**, τότε ανέφερε τη **λύση**.
- Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
- Εφάρμοσε τους **τελεστές μετάβασης** για να βρεις τις **καταστάσεις-παιδιά**.
- Βάλε τις καταστάσεις-παιδιά στο τέλος του μετώπου της αναζήτησης.
- Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
- Πήγαινε στο βήμα 2.

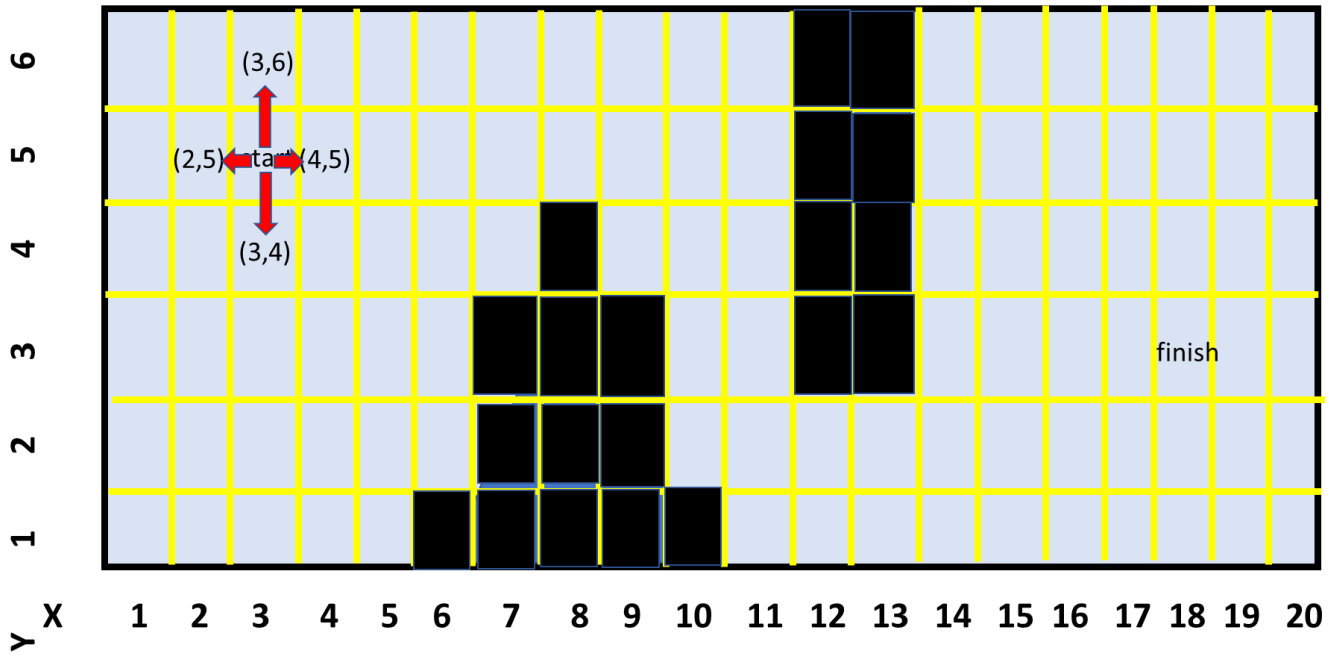
Breadth First Search

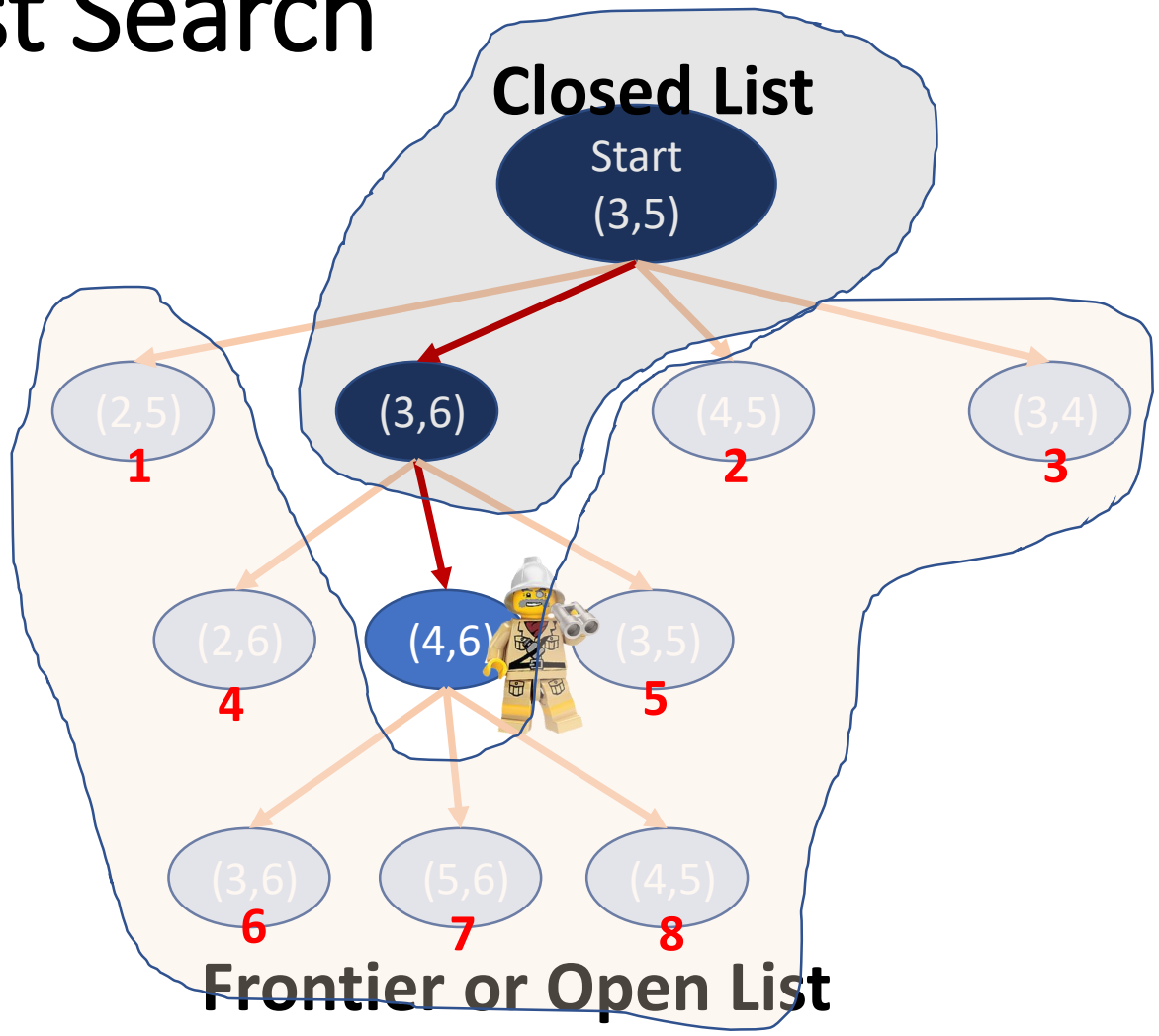




- **Soundness** (Ορθότητα) : All the solutions returned by the algorithm are feasible
- **Completeness** (Πληρότητα): If the problem is solvable the algorithm will return at least one solution
- **Optimality** ("Βελτιστότητα"???): The first solution returned by the algorithm is the optimal ← Optimality Criterion

Breadth First Search





Worst-case complexity:

memory $O(|S|)$

running time $O(b|S|)$

DFS vs BFS

- **Sound**
- **Complete**
- **Not Optimal**
- Time: $O(b^l)$
- Memory: $O(bl)$

- **Sound**
- **Complete**
- **Optimal**
- **Time: $O(b|S|)$**
- **Memory: $O(|S|)$**

Iterative Deepening (IDS)

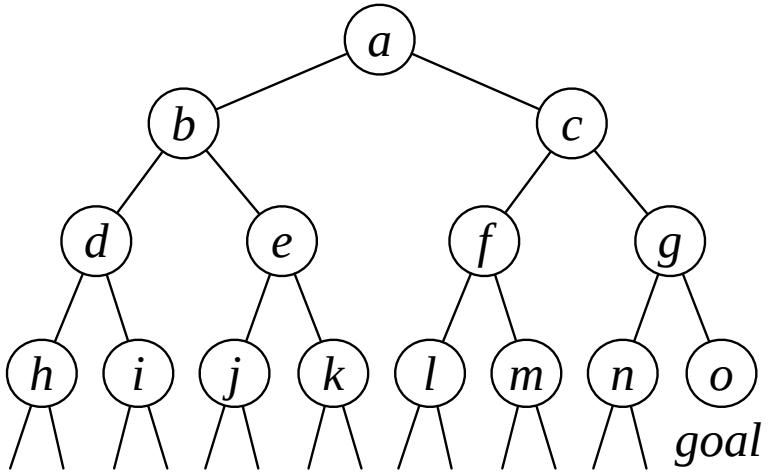
IDS(Σ, s_0, g)
  for k = 1 to ∞ do
    do a depth-first search, backtracking at every node of depth k
    if the search found a solution then return it
    if the search generated no nodes of depth k then return failure

- Example:
  1. Expand *a*
  2. Expand *a,b,c*
  3. Expand *a,b,d,e,c,f,g*
  4. Expand *a,b,d,h,i,e,j,k,c,f,l,m,g,n,o*

  Solution path ⟨*a,c,g,o*⟩

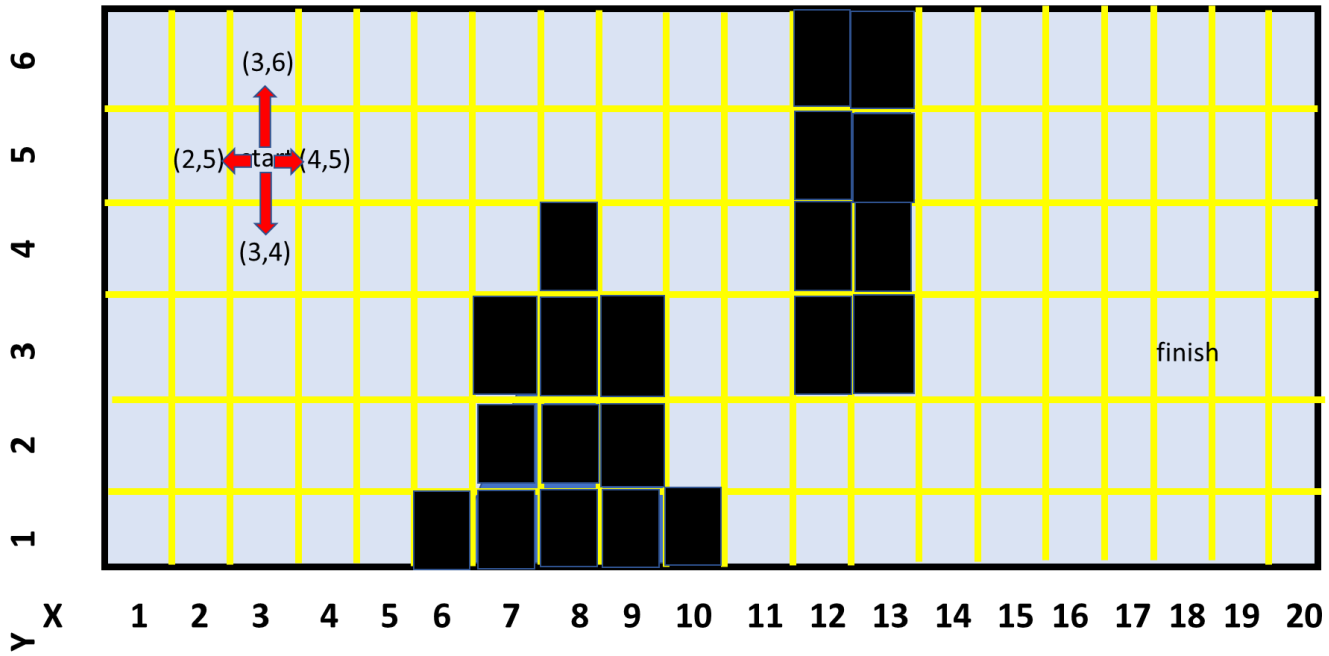
  Total number of node expansions:

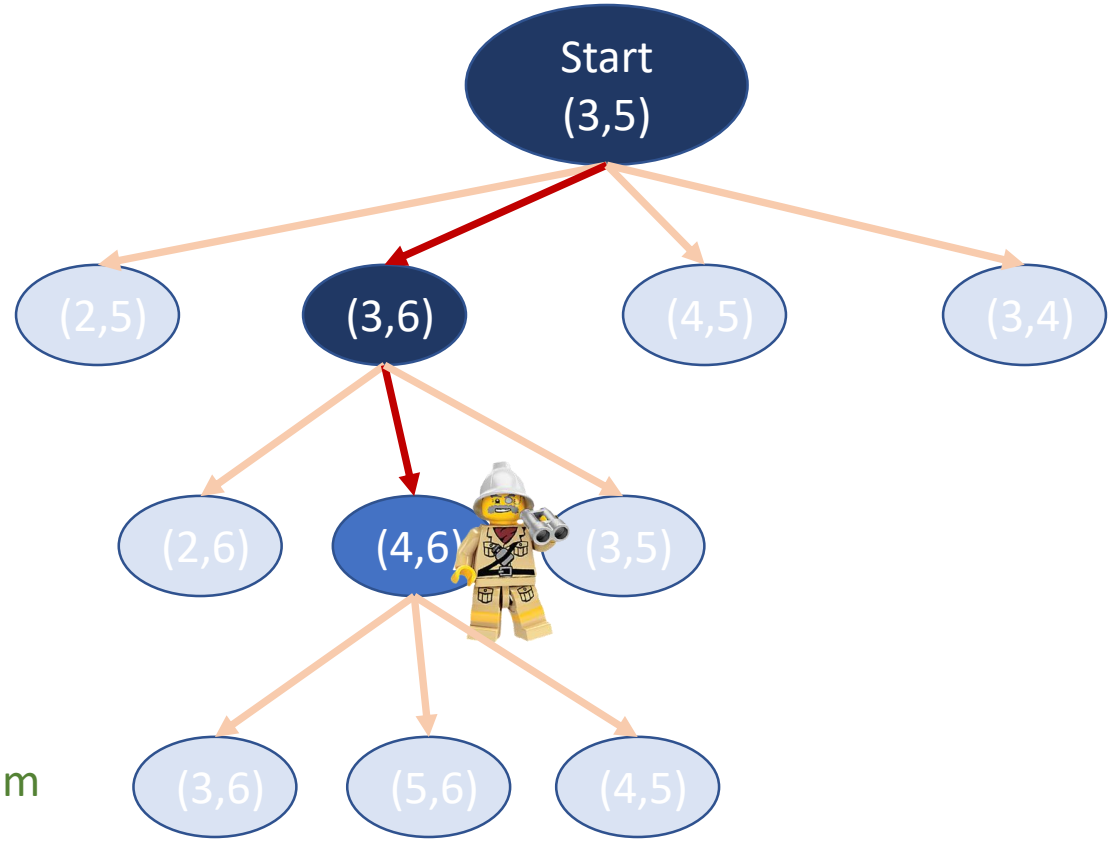
  1+3+7+15 = 26



- If goal is at depth d and branching factor is 2:
  - $\sum_1^d (2^i - 1) = 2^{d+1} - d - 2 = O(2^d)$

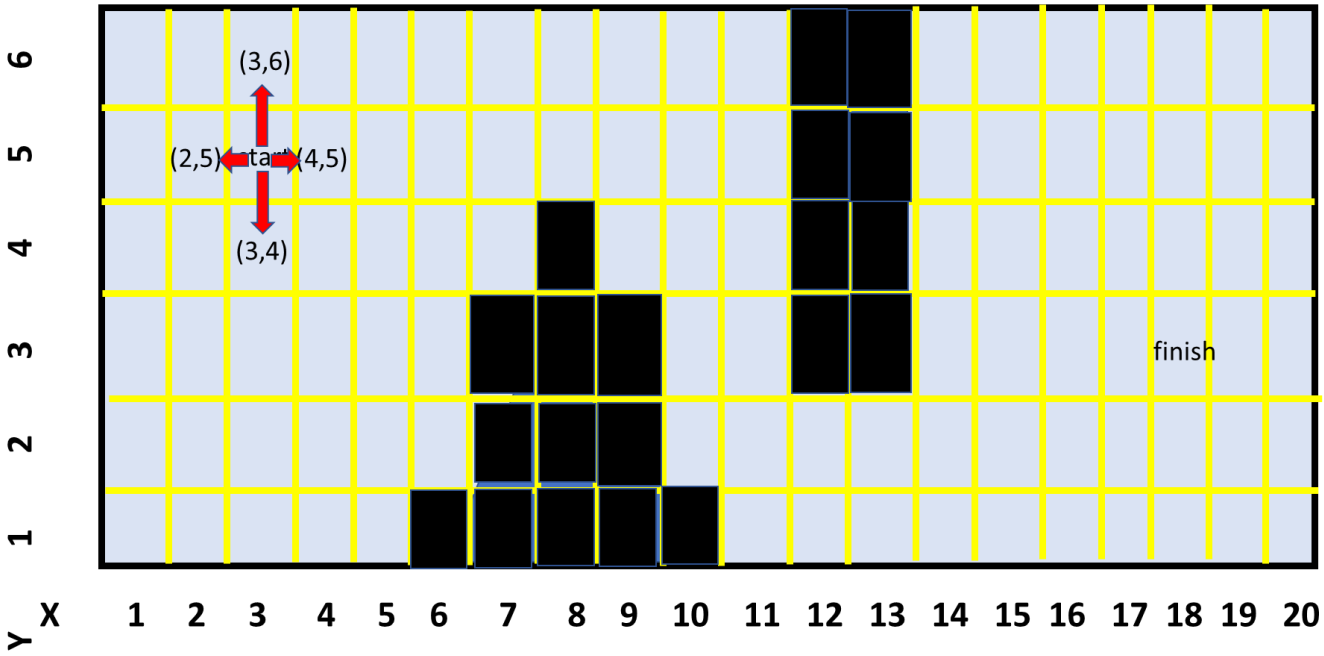
Iterative Deepening

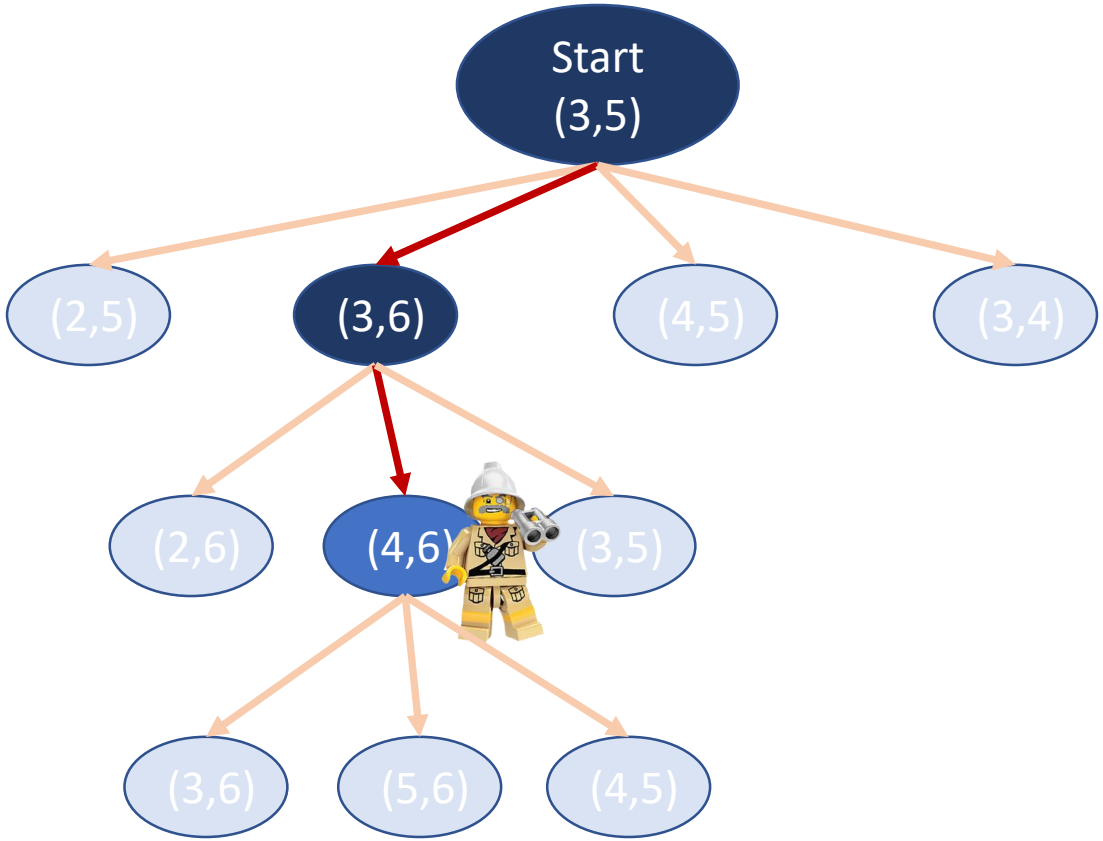




- **Soundness** (Ορθότητα) : All the solutions returned by the algorithm are feasible
- **Completeness** (Πληρότητα): If the problem is solvable the algorithm will return at least one solution
- **Optimality** ("Βελτιστότητα"???): The first solution returned by the algorithm is the optimal ← Optimality Criterion

Iterative Deepening





memory requirement ***O*(*bd*)** vs. $O(b^d)$ for breadth-first search
worst-case running time ***O*(b^d)**, vs. $O(b^l)$ for DFS

b = max branching factor
d = min solution depth if there is one, otherwise max depth of any node

ΑΣΚΗΣΗ

Το παρακάτω σχήμα παρουσιάζει ένα γράφο αναζήτησης, όπου ο αριθμός μέσα σε αγκύλες δίπλα σε κάθε κόμβο αντιστοιχεί στην τιμή μιας ευριστικής συνάρτησης. Οι κόμβοι που έχουν τιμή 0, είναι οι τερματικοί κόμβοι της αναζήτησης. Θεωρώντας ότι οι αλγόριθμοι επιστρέφουν τη πρώτη τερματική κατάσταση που συναντούν να συμπληρώστε το παρακάτω πίνακα:

Αλγόριθμος	Τερματική Κατάσταση	Μήκος μονοπατιού
DFS		
BFS		
ID(2,2)