ITMO UNIVERSITY

Saint Petersburg, Russia

Discrete optimization algorithms

Sequential Search, Depth-First Search, Depth-First Branch-and-Bound, Iterative Deepening Search, Best-First Search

Students:

Anastasiia Belykh, C4132

Mariia Bardina, C4132

Mikhail Sarafanov, C4134

ITMO UNIVERSITY

Content

- **Theory basis**
  - Sequential Search
  - Depth-First Search
  - Depth-First Branch-and-Bound
  - Iterative Deepening Search
  - Best-First Search
- **Practical application**
- **Conclusion**

ITMO more than a UNIVERSITY

ITMO UNIVERSITY

Sequential search works with the list and find an element within one

Table 1. Time complexity of the sequential search

Case	Best case	Worst case
Unordered List		
item is present	O(1)	O(n)
item isn't present	O(n)	O(n)
Ordered List		
item is present	O(1)	O(n)
item isn't present	O(1)	O(n)

Sequential search **useful** when:

- the list has only a few elements
- performing a single search in an unordered list

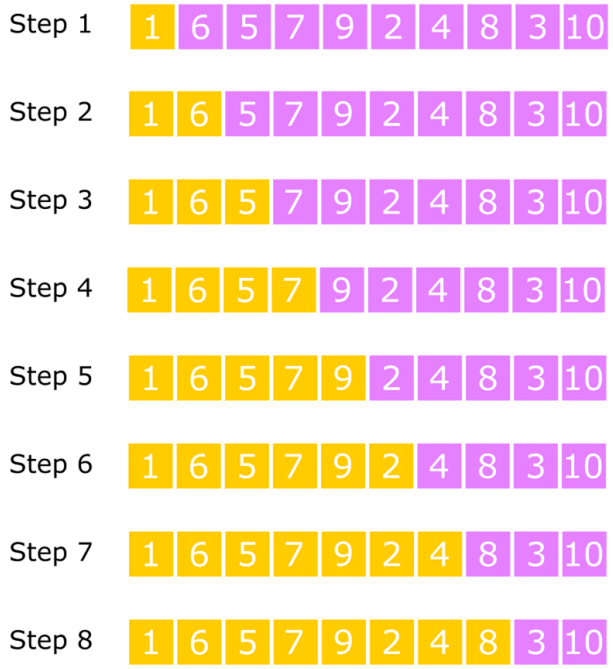

Figure 1. Sequential Search for number "8"

ITMO UNIVERSITY

Depth-First Search

Algorithm is used for traversing or searching tree or graph data structures.

Time complexity: O(|V|+|E|)

Applications:

- testing connectivity
- searching for loops
- finding strongly connected component
- topological sorting
- finding the longest path

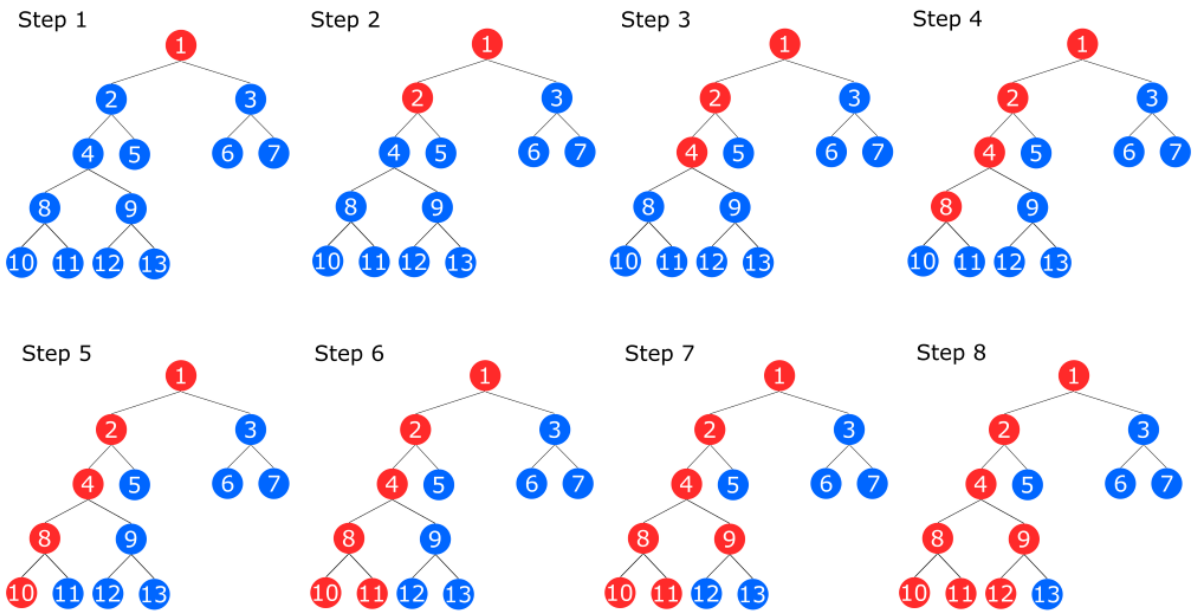

Figure 2. Depth-First Search

ITMO UNIVERSITY

Depth-First Branch-and-Bound

It is used for solving **NP-hard** discrete optimization problems.

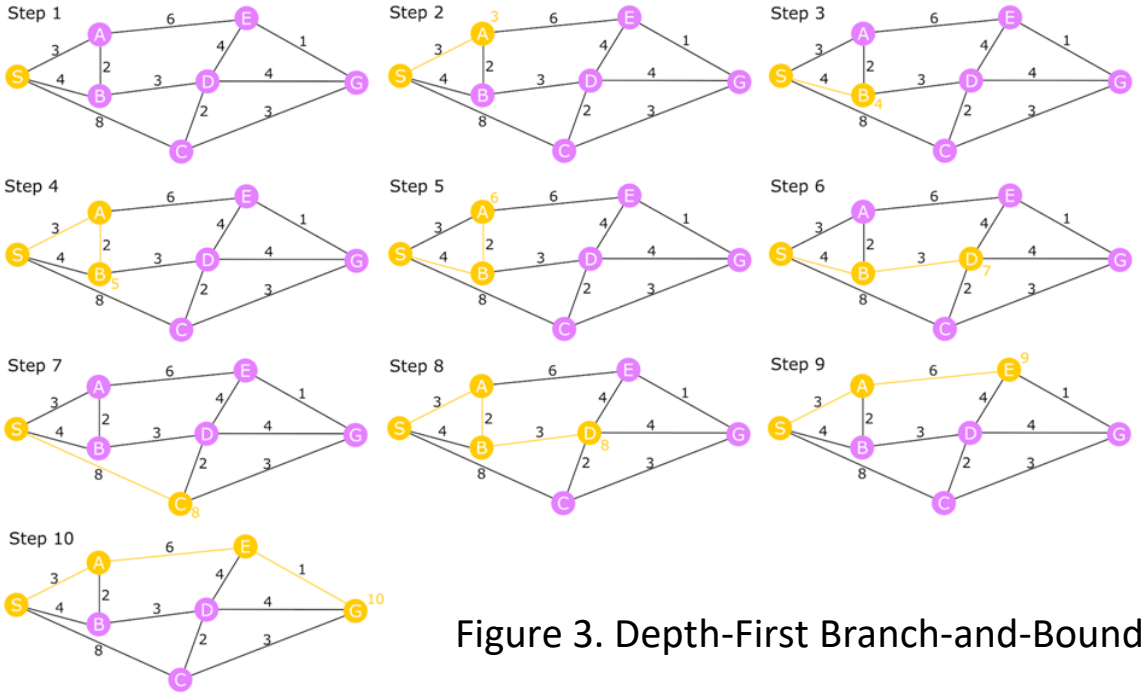

Figure 3. Depth-First Branch-and-Bound

Iterative Deepening Search

Time complexity: $O(b^k)$

Space complexity: O(bk)

Applications:

- chess programs
- best-first heuristic search
- modification the algorithm A*
- finding the shortest path in a graph

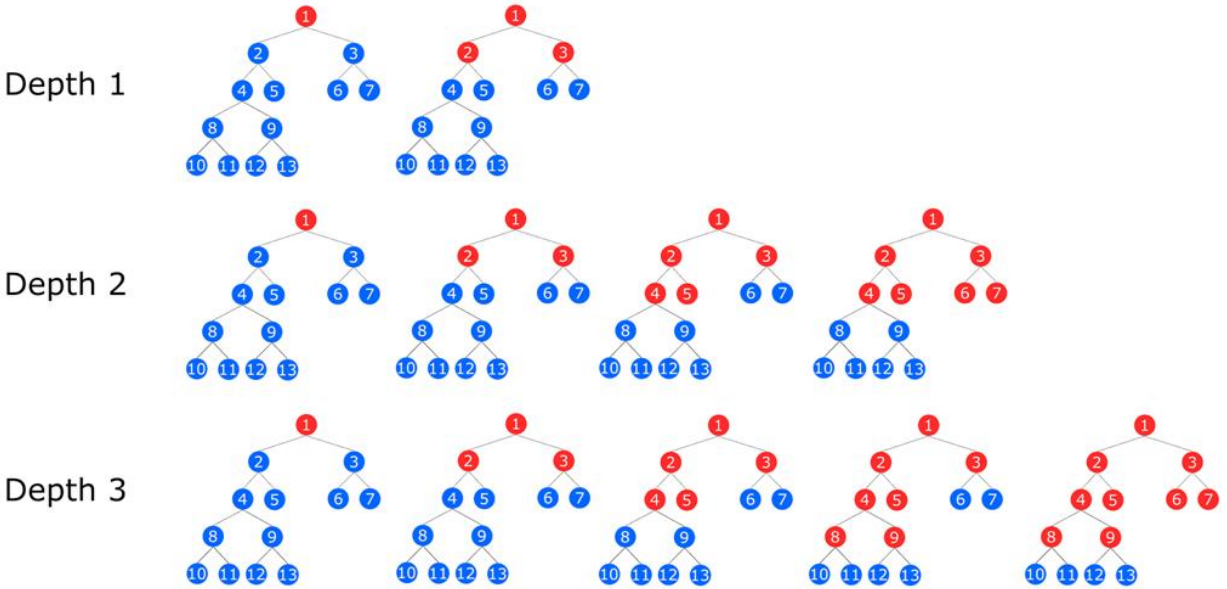

Figure 4. Iterative Deepening Search

Best-First Search

Node is selected for expansion based on evaluation function f(n).

A* algorithm: f(n)= g(n) + h(n)

- **g(n)** - distance traveled
- **h(n)** - remaining distance

Time complexity: O(|E|)

Applications:

- solving the pathfinding problems in computer games

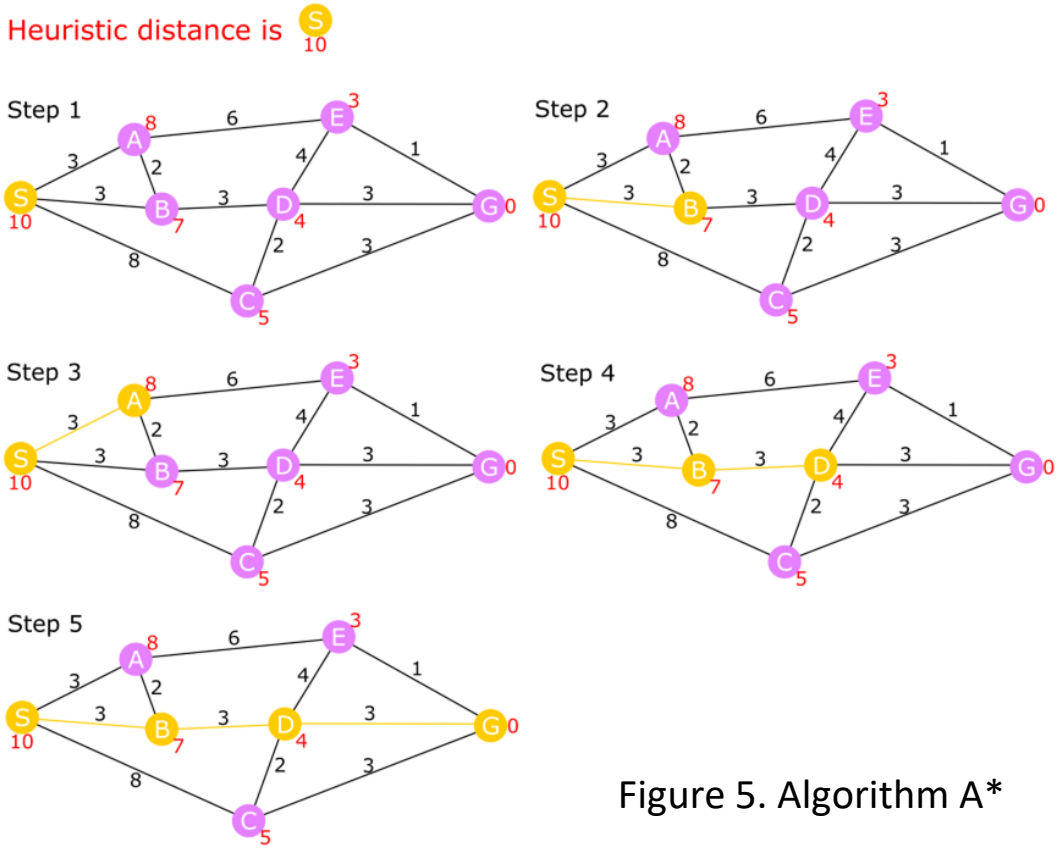

Figure 5. Algorithm A*

Practical application

Graph description

We have undirected weighted graph, where:

- **Nodes** (332) are cities
- **Edges** (2126) are routes between cities
- **Weights** are flight duration
- **Average degree –** 12

Our graph is **connected.**

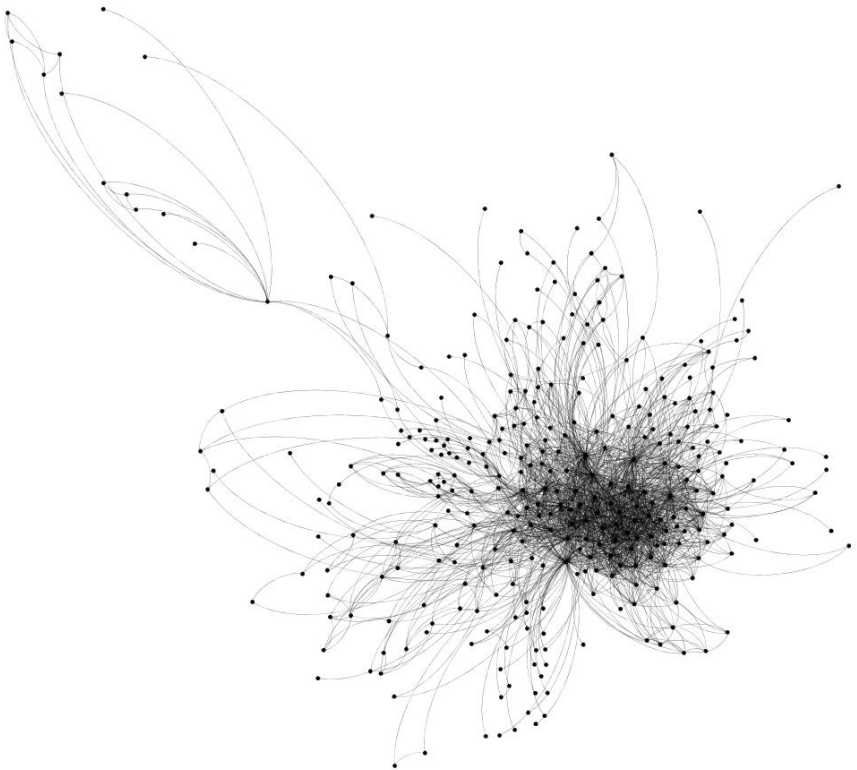

Figure 6. Graph “USAir97"

Experiments results

Task: finding the shortest path

To compare the algorithms by the operating time:

- 1 starting point and 8 end points were chosen
- each algorithm was run 10 times

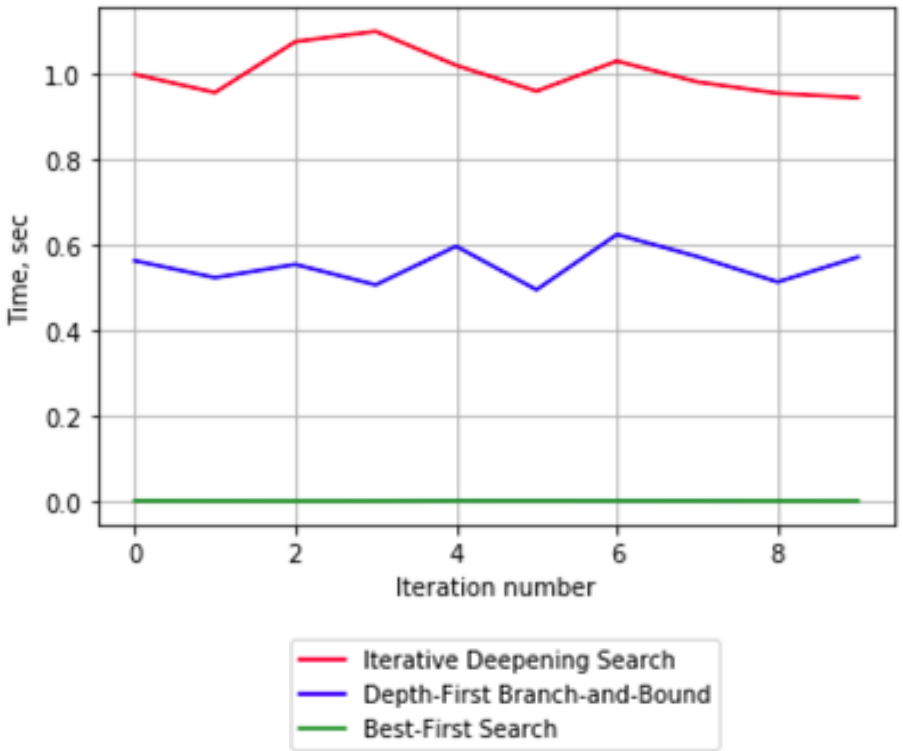

Figure 7. Execution time during iterations

Conclusion

Best-First Search algorithm is proved to be most effective in terms of runtime

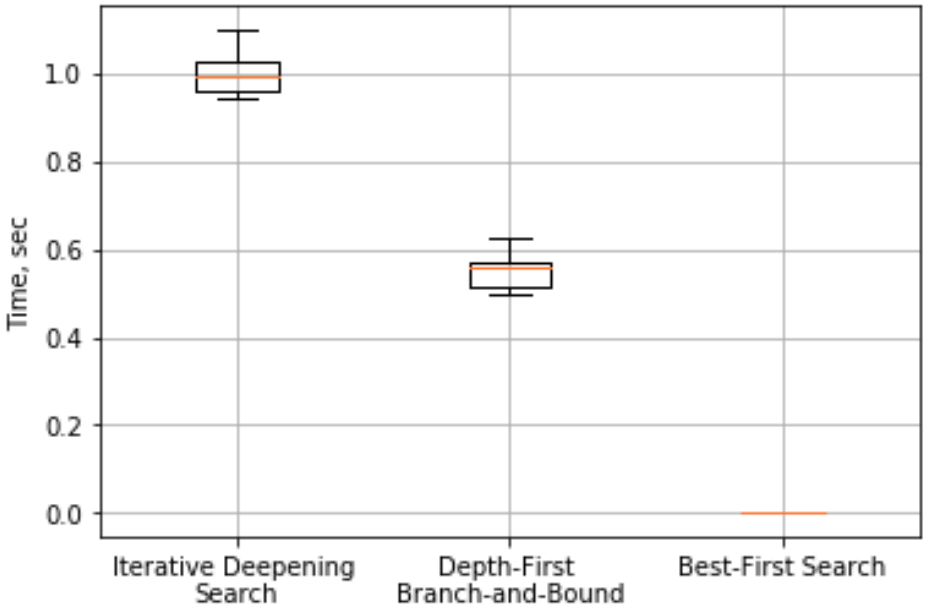

Figure 8. Boxplot presentation of the results

Thanks for attention!

www.ifmo.ru

ITMOre than a UNIVERSITY