CS 2050
Computer Science II

Lesson 07

METROPOLITAN STATE UNIVERSITY℠
OF DENVER
LIVES TRANSFORMED

Agenda

- Exhaustive Search
- Backtracking

Exhaustive Search

- Also called “brute-force” search
- Problem-solving technique that systematically enumerates all possible candidates of a solution to a problem, checking if each of them solves the problem

Exhaustive Search

- Example:
  - “Search for all of the combinations of letters with a given size”

Exhaustive Search

- Example - queue solution:
    - Ask for the size (parameter “n”)
    - Create a queue of String objects
    - Push all letters of the alphabet onto the queue
    - Loop until the queue is empty

Exhaustive Search

- Example - queue solution:
  - Loop until the queue is empty:
    - Pop a String from the queue
    - IF size of the String is “n”, print the String
    - ELSE, push back all of the combinations of the String with each individual letters of the alphabet

Exhaustive Search

- Example - stack solution:
  - Ask for the size (parameter “n”)
  - Create a stack of String objects
  - Push all letters of the alphabet onto the stack
  - Loop until the stack is empty

Exhaustive Search

- Example - stack solution:
  - Loop until the stack is empty:
    - Pop a String from the stack
    - IF size of the String is “n”, print the String
    - ELSE, push back all of the combinations of the String with each individual letters of the alphabet

Exhaustive Search

- Conclusions:
    - Stacks enable a Depth-first Search (DFS)
    - Queues enable a Breadth-first Search (BFS)

Exhaustive Search

- Conclusions:
  - A stack implementation will implement a search using considerably less memory in comparison to a queue implementation

Exhaustive Search

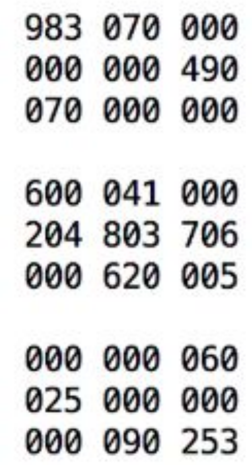
- Conclusions:
  - A stack implementation makes it possible to backtrack to the previous computation

Backtracking

- The exhaustive search involves generating and verifying ALL possible solutions to a problem
- Backtracking works more efficiently because it checks partial solutions and backtracks (i.e., retraces one step) if the partial solution is not worth continuing

Backtracking

- Example:
  - Solve a Sudoku puzzle!



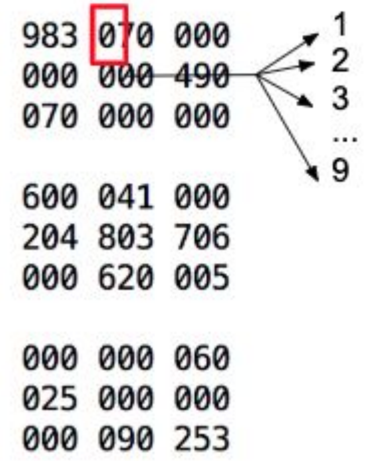
```
983 070 000
000 000 490
070 000 000

600 041 000
204 803 706
000 620 005

000 000 060
025 000 000
000 090 253
```

Backtracking

- Example:
  - Solve a Sudoku puzzle!



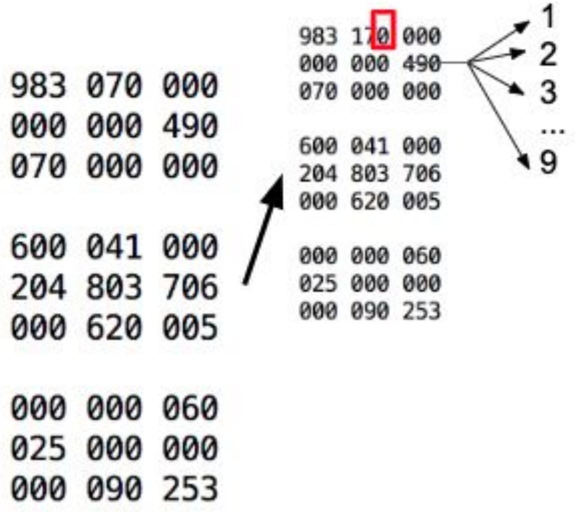
```
983 070 000
000 000 490
070 000 000

600 041 000
204 803 706
000 620 005

000 000 060
025 000 000
000 090 253
```

Backtracking

- Example:
  - Solve a Sudoku puzzle!



```
983 070 000
000 000 490
070 000 000

600 041 000
204 803 706
000 620 005

000 000 060
025 000 000
000 090 253
```

```
983 170 000
000 000 490
070 000 000

600 041 000
204 803 706
000 620 005

000 000 060
025 000 000
000 090 253
```