Força Bruta - BackTracking

Prof. Leandro Israel Pinto

Força Bruta

- Abordagem mais direta para resolver um problema
- Se baseia na tentativa de encontrar uma solução simplesmente testando todas as possibilidades
- Depende do poder computacional e não da inteligência na modelagem do algoritmo

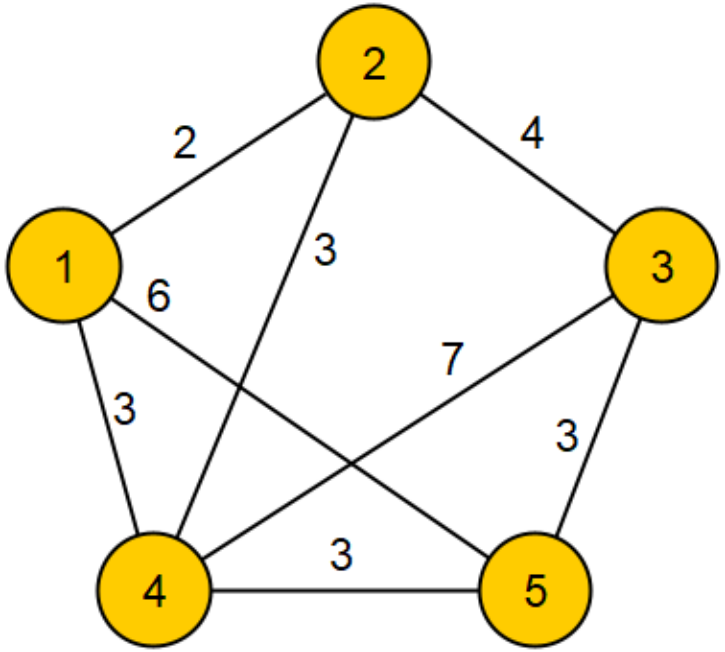
Força Bruta

Exemplos:

- Busca Sequecial
- Multiplicação de Matrizes
- Bubble Sort

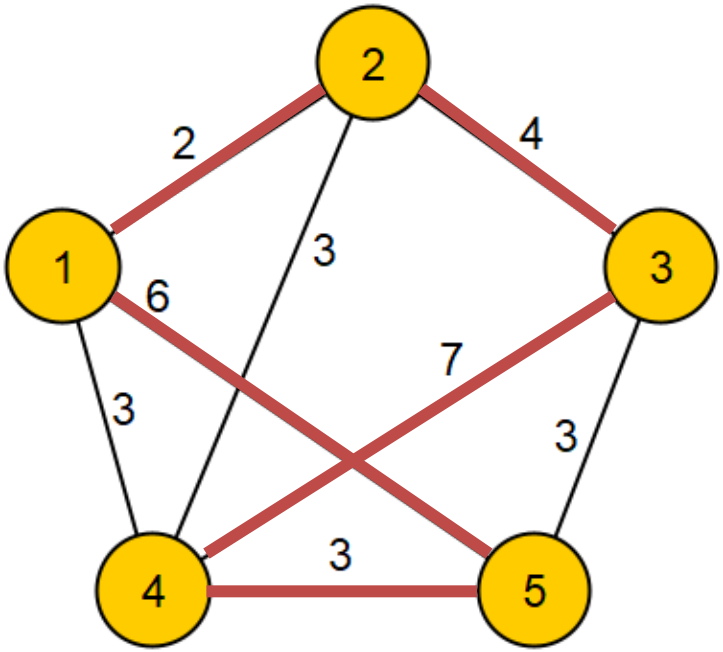
Força Bruta

- Problema do caixeiro viajante
  - Passar por cada cidade uma única vez e voltar à origem, considerando o custo mínimo



Força Bruta

- Problema do caixeiro viajante
  - Passar por cada cidade uma única vez e voltar à origem, considerando o custo mínimo
  - [1, 2, 3, 4, 5, 1] = Custo 22



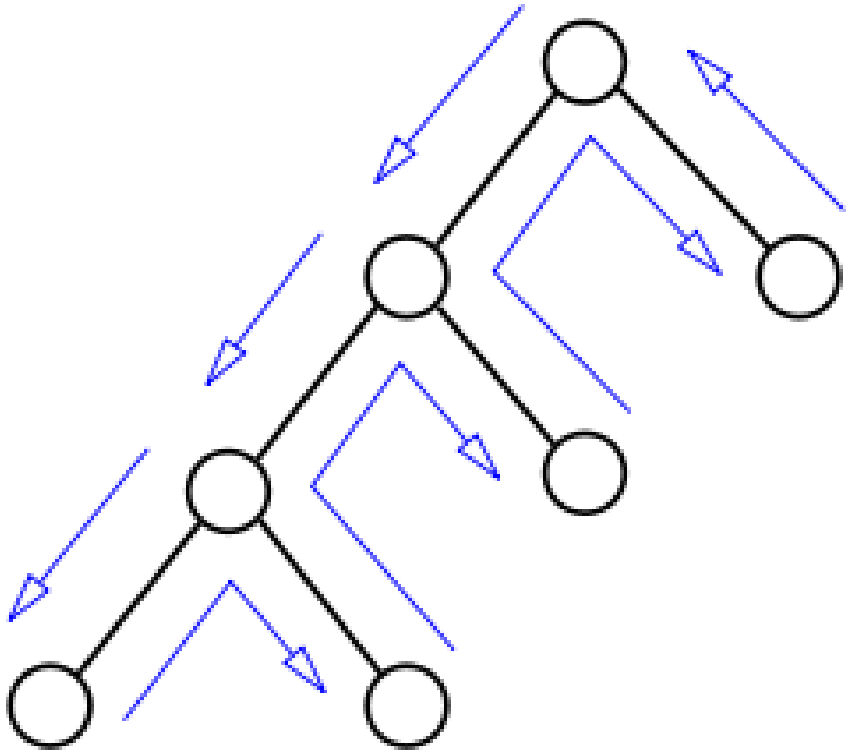
Força Bruta

- Existem muitos problemas que são difíceis de resolver
  - Difícil de propor um algoritmo eficiente
- Existem duas técnicas de projeto de algoritmo para tentar diminuir o espaço de busca
  - Mas, no pior caso, ainda enfrentam a explosão exponencial da busca exaustiva

- Para melhorar a busca exaustiva há duas técnicas:
  - Backtracking
    - Recuar e Retroceder
  - Branch-and-bound
    - Ramificar e limitar
- Baseadas na construção de uma árvore de estados
  - Os nós refletem uma escolha feita em direção a uma solução

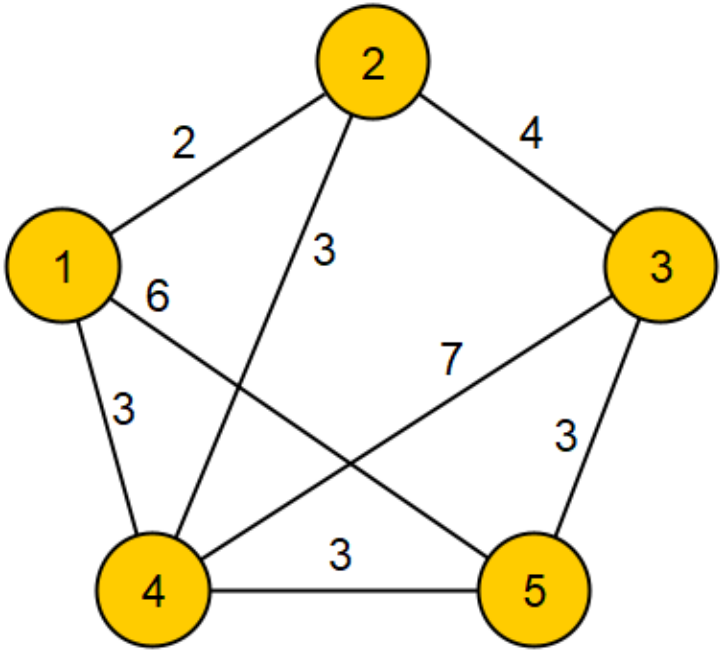
Backtracking

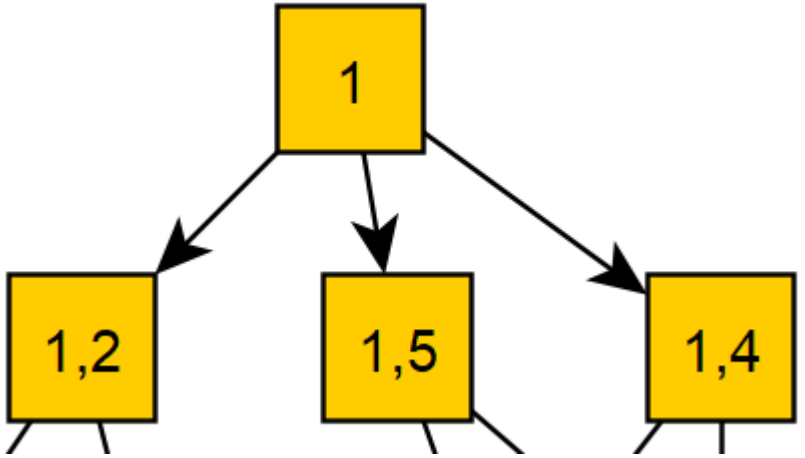
- Dado um estado inicial, a ca passo em direção a solução cria-se um novo nó
  - Retrocede sempre que a sequência de passos não atingiu uma solução
  - Ou, quando se deseja continuar procurando por uma solução melhor
    - Branch-and-Bound



Backtracking

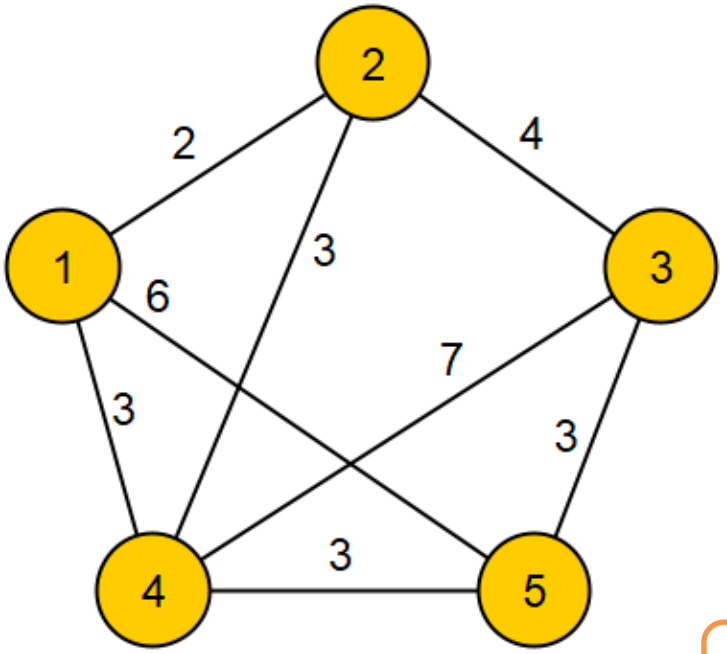
- Exemplo: Iniciando em 1, são possíveis 3 passos. De 1 para 2 de 1 para 5 ou de 1 para 4

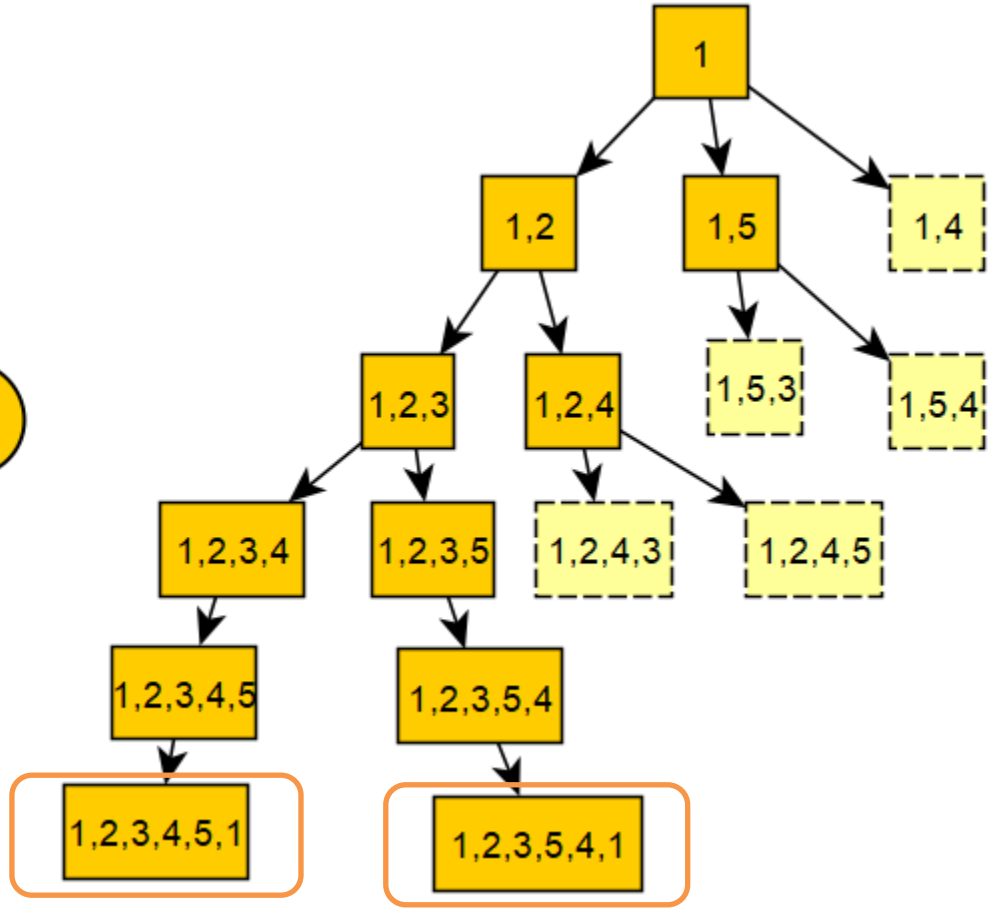




Backtracking

- Exercício: encontre uma solução a partir do nó [1,4] e um caminho na árvore que não leve a uma solução.





Exercício

- Listar outros 3 problemas que podem ser resolvidos por backtracking
- Para cada um
  - Apresentar o que cada nó e/ou aresta da árvore representa
  - Mostrar a árvore gerada para resolver um problema pequeno

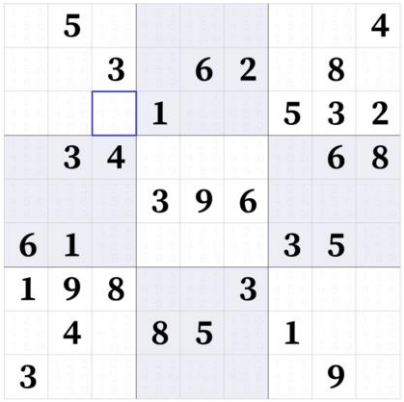
Isso vale nota

Backtracking em Jogos

- Algoritmos backtracking podem ser utilizados para resolver vários jogos
- Caracterizados pela necessidade de explorar uma faixa de possibilidades a cada ponto de escolha
- BT simplifica os dados. A cada nível de recursão considera-se uma escolha. O histórico de quais escolhas foram feitas é mantida na pilha de execução

Backtracking em Jogos

- Jogos solitários
  - Testa todas as possibilidades
  - Labirinto, Torre de Hanoi, Sudoku, Resta 1, Arranjo
  - E muitos outros: https://www.mathsisfun.com/games/puzzle-games.html

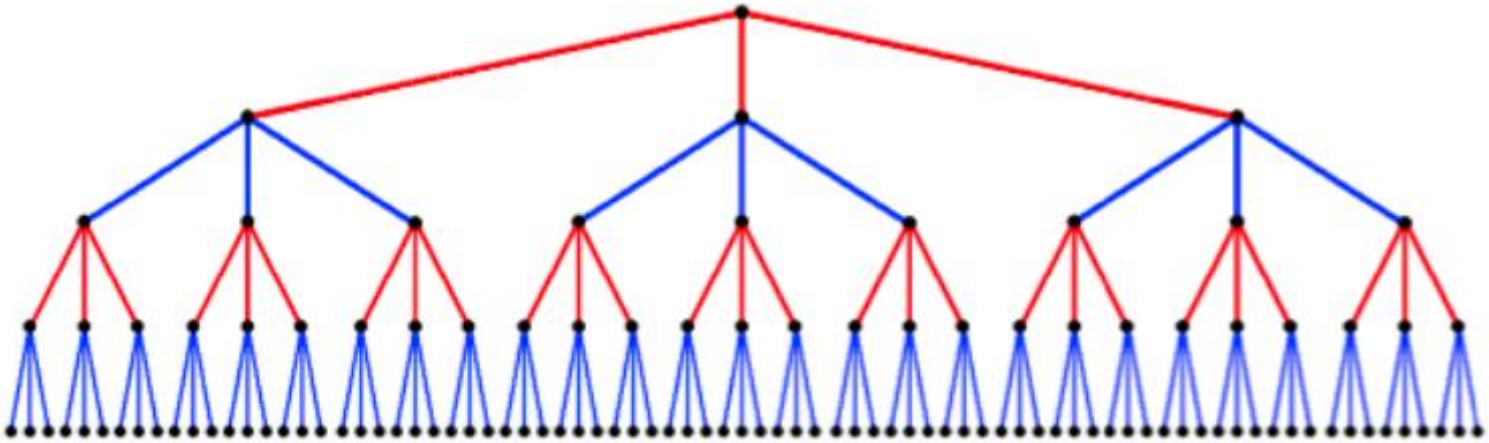


Backtracking em Jogos

- Em jogos de duplas
  - O algoritmo backtracking é utilizado para gerar todas as possíveis jogadas do computador e da pessoa;
  - O objetivo é seguir pelas jogadas que deixem o oponente com menos opções

Backtracking em Jogos

- Claude Shannon observou que a maioria dos jogos de 2 jogadores tem a mesma forma básica.
- Cada nível da árvore representa a jogada de um dos jogadores.



Exemplo

- O jogo Nim consiste numa pilha de moedas.
- Cada jogador pode pegar até 3.
- Aquele que ficar com a última perde.

```
Tem 11 moedas. Quantas voce quer? :3
Vou pegar 3.
Tem 5 moedas. Quantas voce quer? :1
Vou pegar 3.
Sobrou 1.
** Voce Perdeu! **
```

Exercício

- Implemente um jogo Nim que permita o jogador jogar contra o PC.
  - Analise a árvore de recursão, mostre o parâmetro e o retorno em cada nó
  - Descubra como o computador sabe a melhor jogada a se fazer

```
int findBestMove2(int nCoins){
    int nTaken;
    int limit = (nCoins < MAX_MOVE) ? nCoins : MAX_MOVE;
    for(nTaken = 1; nTaken <= limit; nTaken++){
        if(nCoins-nTaken == 1 ) return nTaken;
        if(findBestMove2(nCoins-nTaken)==-1) return nTaken;

    }
    return -1;
}
```