теми за контролно 1

1. **Изкуствен интелект - въведение (тест на Тюринг)**
2. **Интелигентни агенти**
3. **Решаване на проблеми, чрез търсене, Неинформирано търсене**
4. **Информирано търсене**
5. **Локално търсещи алгоритми**
   1. **Hill climbing**
   2. **Simulated annealing**
6. **Genetic algorithms**
7. **CSP**
8. **Игри**
9. **Knowledge representation**
10. **Expert systems**

* характеристики на алгоритъм
  + пълен (completeness) - дали винаги открива решение, когато такова съществува
  + оптимален (optimality) – пълен алгоритъм, който при повече от едно решение открива решението с най-малка цена
  + сложност по време – максималният брой възли, който алгоритъма обхожда
  + сложност по памет – максималният брой възли, които алгоритъмът пази в паметта в даден момент

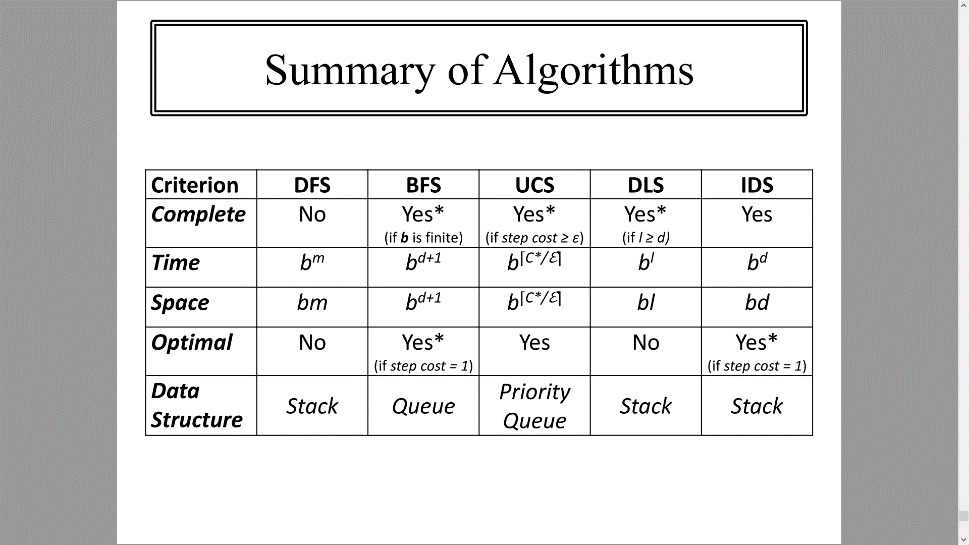
**b -** (branching factor) - максималният брой деца, които може да има дадено състояние

**d** – дълбочина на решението с най-малка цена

**m** – максимална дълбочина на пространството от състояния

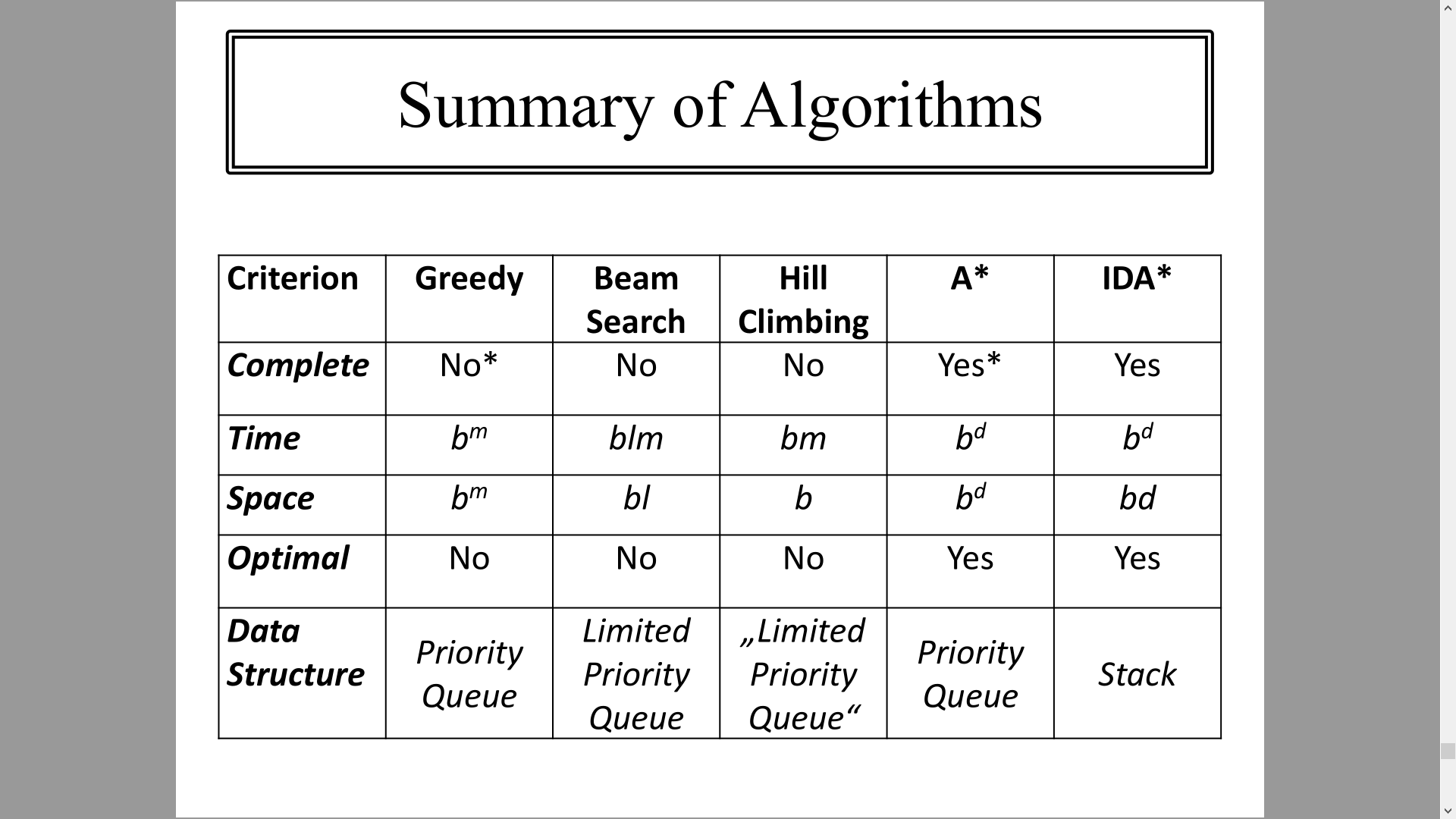
Подробно

1. **Изкуствен интелект - въведение (тест на Тюринг)**
   1. тестът на Тюринг представлява експеримент, при който човек трябва да комуникира с компютърна система. Общуването продължава някакъв период от време и за да бъде тестът успешен, човекът не трябва да може разграничи дали това, с което си е говорил е било машина или човек.
2. **Интелигентни агенти**
3. **Решаване на проблеми, чрез търсене, Неинформирано търсене**

****

* 1. DFS – използва стек за фронта
  2. BFS – използва опашка за фронта
  3. Uniform Cost Search – използва приоритетна опашка за фронта
  4. Depth Limited Search – DFS с граница на дълбочината
  5. Iterative Deepening Search – пускане на DLS с нарастваща дълбочина, комбинация на DFS и BFS(експоненциална сложност по време, линейна памет, оптималност)

1. **Информирано търсене**



* 1. Best-first search
     1. Greedy Best First Search – фронтът е приоритетна опашка, като оценката на възлите е само стойността на тяхната евристична функция
     2. Beam Search – държи най-добрите l-възела, според тяхната евристика в опашка
     3. Hill Climbing – държи само 1 възел, оценява отново само на база евристиката
  2. A\*
     1. A\* - използва оценъчна функция f = g + h, g e дължината на пътя до възела, h е евристична функция
     2. Memory-bound A\* - А\* където опашката е с лимит
     3. Iterative Deepening Search – използва оценъчната функция f и изпълнява Iterative Deepening Search

1. **Локално търсещи алгоритми**
   1. Hill climbing
   2. Simulated annealing
2. Genetic algorithms
3. **CSP**
4. **Игри**
   1. Minimax
   2. Minimax with Alpha-beta prunning
5. **Knowledge representation**
6. **Expert systems**