* характеристики на алгоритъм
  + пълен (completeness) - дали винаги открива решение, когато такова съществува
  + оптимален (optimality) – дали открива решението с най-малка цена
  + сложност по време
  + сложност по памет

**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

1. **неинформирано търсене**
2. **информирано търсене**
   1. какъв е най-големият проблем на А\*? – паметта е експоненциална (b^d)
   2. какво трябва да изпълнява една евристика, така че А\* са е оптимален
3. **задачи за удовлетворяване на изискванията**
   1. какви видове ограничения имаме при CSP
      1. унарни - такива, които включват една променлива
      2. бинарни - такива, които включват две променливи
      3. от по-висок ред - такива, които включват повече от две променливи
      4. преференции(soft contraints) – свързани са с цената за присвояване на дадена променлива
   2. какви видове алгоритми се използват за решаване на CSP?
      1. backtracking – DFS с подобрения
      2. constraint propagation – arc consistency
      3. local search – min conflicts(hill climbing)
   3. кой алгоритъм стои за min conflicts – hill climbing
   4. какъв проблем има hill climbing – може да „заседне“ в рамена от графиката(места, които са flat ?или в някакъв локален екстремум?)
   5. кой подход може да се използва за преодоляване на проблемите на hill climbing(local search)
      1. random restart – пускаме алгоритъма отначало, като рандомизираме началното състояния
      2. simulated annealing, при него на стъпките се задават вероятности, които с времето намаляват, това предотвратява цикленето на едно място
4. **генетични алгоритми**
5. **игри**