Shortest path

1. definition

given a (directed) graph $G = (V, E)$, find the shortest path from s to t.

2. Dijkstra's algorithm

- **1. Basic:**
  - single-source shortest path problem
  - Only for non-negative weights ($w_i \geq 0$)
  - Time: $O((|V| + |E|) \log |V|)$. (Binary heap)以及剩下的... Disclaimer: 在CS101**大题**中，要求分析复杂度只需要考虑Binary Heap实现的Dijkstra就可以，不要求考虑Fibonacci Heap实现的$O(|E| + |V| \log |V|)$
- **2. hw10, Q5**

3. Bellman-Ford algorithm

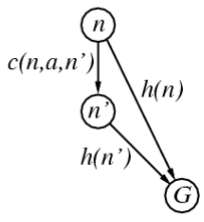
- **1. Basic:**
  - Time : $O(|V| \cdot |E|)$
  - Could have negative weights
- **2. Some questions**
  - If only run i iterations instead of $|V| - 1$ iterations, what does the $\texttt{dist}[v]$ mean now?
  - How to detect negative cycles?
  - How to run faster?

4. A-star algorithm

- **1. heuristic function** h
  - $w(v) = d(a, u) + d(u, v) + h(v, z)$, where a is the initial vertex, and z is the destination vertex. (我不知道要不要讲一下A*是怎么跑的)
  - **Admissible** : $h(u, v)$ is admissible if $h(u, v) \leq d(u, v)$.
    - If h is admissible, then A^* **TREE-SEARCH** is optimal
  - **Consistent** :

A heuristic is consistent if for every node *n*, every successor *n'* of *n* generated by any action *a*, we have

$$h(n) \leq c(n,a,n') + h(n')$$



If *h* is consistent, we have

w(n')	= d(n') + h(n')	(by def.)
	= d(n) + c(n,a,n') + h(n')	(d(n')=d(n)+c(n,a,n'))
	≥ d(n) + h(n) = w(n)	(consistency)
w(n')	≥ w(n)	

i.e., *w(n)* is non-decreasing along any path.

It's the triangle inequality !

（可能需要更好的图，这个有点怪）

  - If h is consistent, then A^* **GRAPH-SEARCH** is optimal
  - If h is consistent, then we can imply h is admissible.

    - Question: 有没有可能h没有什么好的性质导致A^*的搜索结果不是最优呢? 显然是有的，但使用heuristic的方法去搜索通常会比普通的暴力更加有效。
- **2. Tree-Search and Graph-Search (我怀疑太多了，讲这个讲不完)**