11-D1

图应用

Dijkstra算法：最短路径

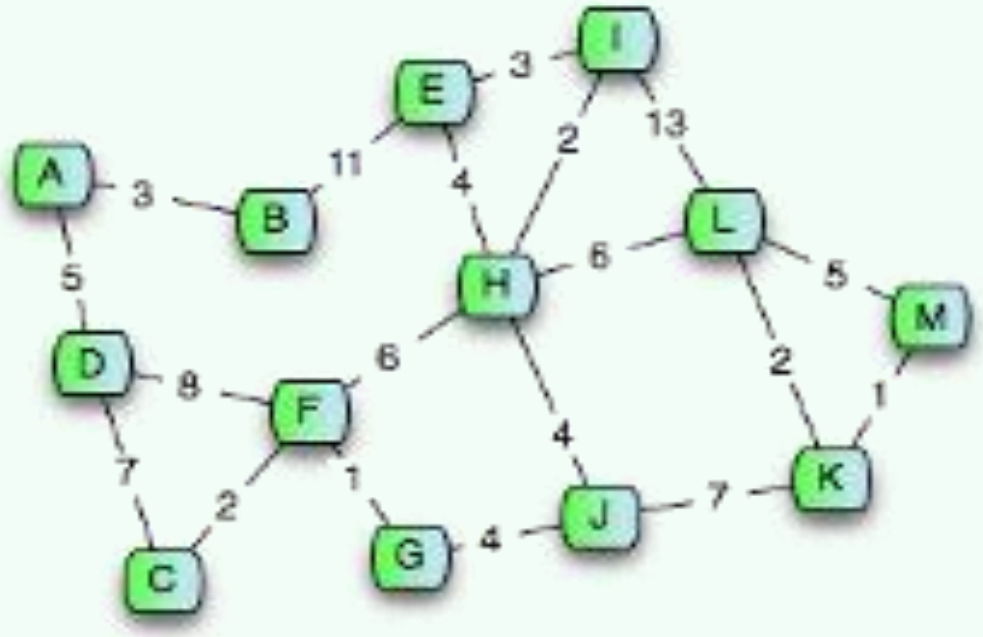
The hurricane seeks the shortest road by the no-road, and suddenly ends its search in the nowhere.

邓俊辉
deng@tsinghua.edu.cn

问题 + 应用

❖ 给定：连通有向图G及其中的顶点u和v

找到：从u到v的最短路径及其长度

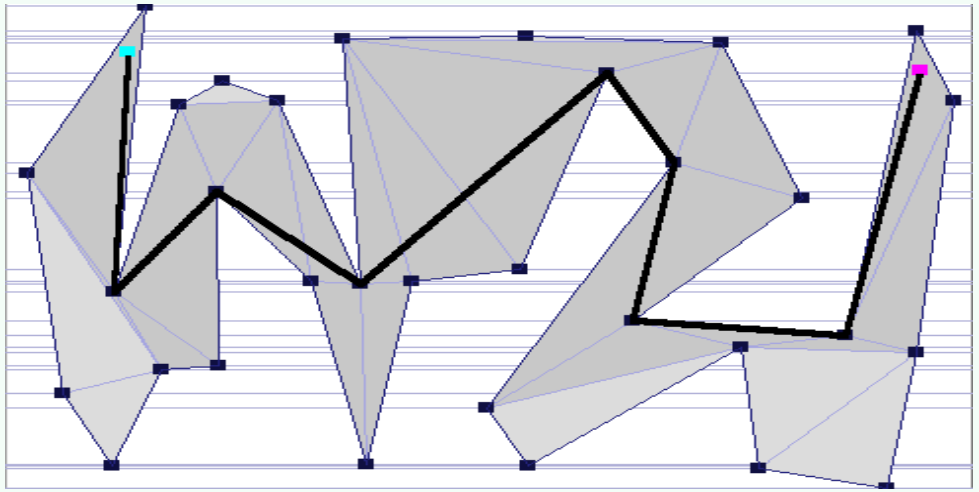


❖ 旅游者：最经济的出行路线

路由器：最快地将数据包传送到目标位置

路径规划：多边形区域内的自主机器人

.....

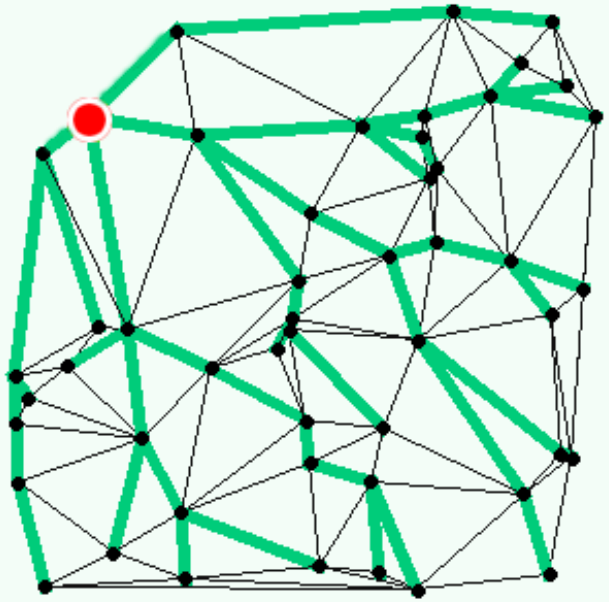


问题分类

❖ SSSP: Single-Source Shortest Path

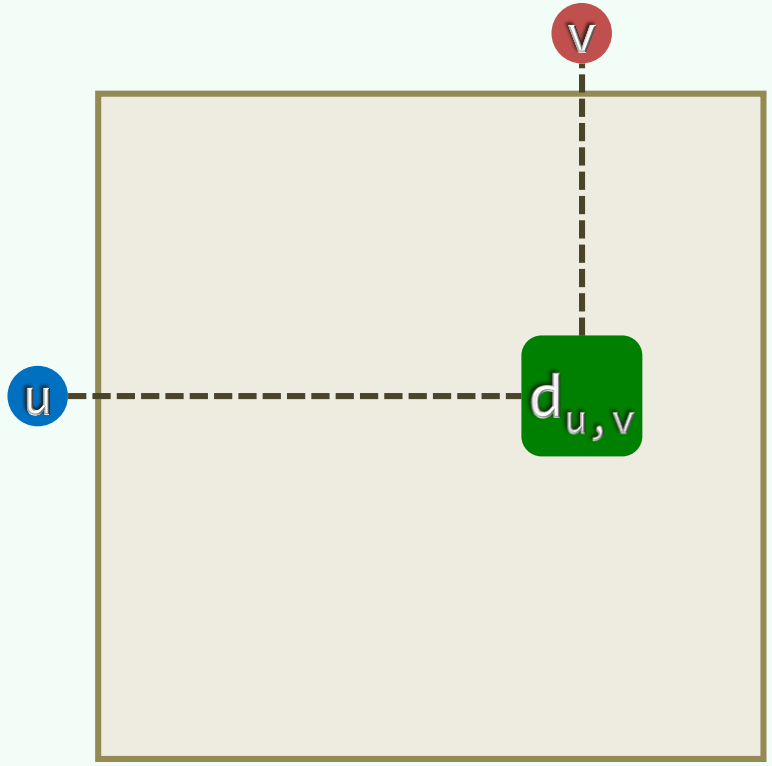
给定顶点s，计算s到

其余各个顶点的最短路径及长度

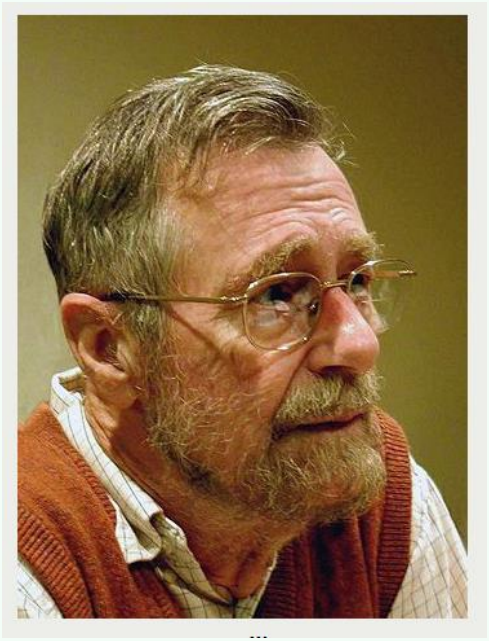


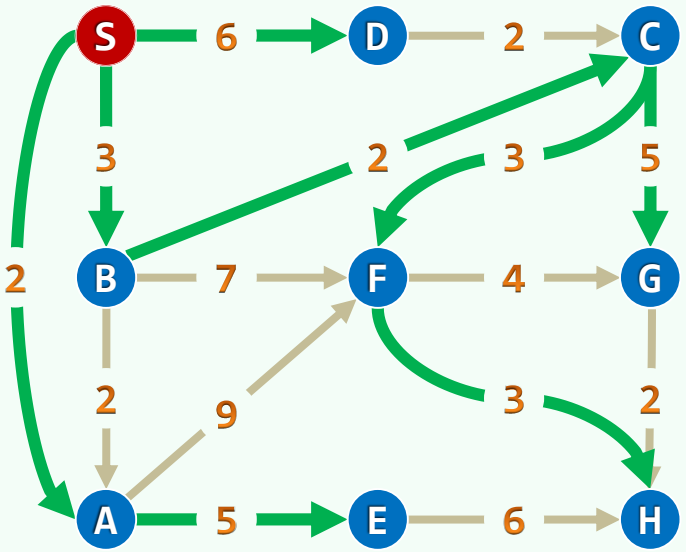
❖ APSP: All-Pairs Shortest Path

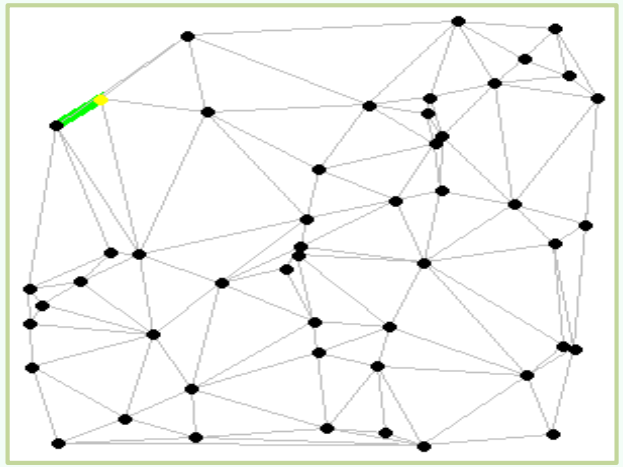
找出每对顶点u和v之间的最短路径及长度



E. W. Dijkstra







1972 ACM Turing Award Lecture

[Extract from the Turing Award Citation read by M.D. McIlroy, chairman of the ACM Turing Award Committee, at the presentation of this lecture on August 14, 1972, at the ACM Annual Conference in Boston.]

The working vocabulary of programmers everywhere is studded with words originated or forcefully promulgated by E.W. Dijkstra—display, deadly embrace, semaphore, go-to-less programming, structured programming. But his influence on programming is more pervasive than any glossary can possibly indicate. The precious gift that this Turing Award acknowledges is Dijkstra's *style*: his approach to programming as a high, intellectual challenge; his eloquent insistence and practical demonstration that programs should be composed correctly, not just debugged into correctness; and his illuminating perception of problems at the foundations of program design. He has published about a dozen papers, both technical and reflective, among which are especially to be noted his philosophical addresses at IFIP,[1] his already classic papers on cooperating sequential processes,[2] and his memorable indictment of the go-to statement.[3] An influential series of letters by Dijkstra have recently surfaced as a polished monograph on the art of composing programs.[4]

We have come to value good programs in much the same way as we value good literature. And at the center of this movement, creating and reflecting patterns no less beautiful than useful, stands E.W. Dijkstra.

The Humble Programmer

by Edsger W. Dijkstra

As a result of a long sequence of coincidences I entered the programming profession officially on the first spring morning of 1952, and as far as I have been able to trace, I was the first Dutchman to do so in my country. In retrospect the most amazing thing is the slowness with which, at least in my part of the world, the programming profession emerged, a slowness which is now hard to believe. But I am grateful for two vivid recollections from that period that establish that slowness beyond any doubt.

After having programmed for some three years, I had a discussion with van Wijngaarden, who was then my boss at the Mathematical Centre in Amsterdam—a discussion for which I shall remain grateful to him as long as I live. The point was that I was supposed to study theoretical physics at the University of Leiden simultaneously, and as I found the two activities harder and harder to combine, I had to make up my mind, either to stop programming and become a real, respectable theoretical physicist, or to carry my study of physics to a formal completion only, with a minimum of effort, and to become . . . , yes what? A programmer? But was that a respectable profession? After all, what was programming? Where was the sound body of knowledge that could sup-

1,2,3,4 Footnotes are on page 866.