

**Lab Report**

**实验报告**

|  |  |
| --- | --- |
| **Course**: | Discrete mathematics |
| **Semester**: | 1st semester of the academic year **2023-2024** |
| **Major**: | Software Engineering |
| **Class**: | 2022 |
| **Student Name**: | 朱昊 |
| **Student ID:** | 222022321062008 |
| **Teacher:** | 詹成 |

**School of Computer and Information Science**

足球是世界三大球类运动之一，也是目前全世界范围内最受欢迎的运动，深受各个年龄段人的喜爱。在绿茵场上，22名球员运用技巧、速度和团队合作，力争将足球踢入对方球门，这是一场技艺与策略的完美融合。而在电子游戏领域，EA旗下的FIFA系列（如下图所示）为玩家提供了身临其境的足球体验，让他们能够亲身感受比赛的激动与紧张。



然而，在这个虚拟的足球世界中，虽然球员们看似自由奔跑，但背后却隐藏着复杂的计算和策略。在游戏开发中，寻找最佳策略成为关键，尤其是关于球员移动的路径选择。同样，游玩的玩家也需要在变幻莫测的球场中迅速寻找出最优策略，这便引出了离散数学中的一个经典问题：最短路径。在FIFA游戏中，玩家需要控制球员通过精准的传球和移动来攻破对方防线，而这需要找到最短的路径，以确保球队能够高效地推进并最终得分。因此，游戏中的最短路径问题成为了一项引人入胜且具挑战性的计算任务，为玩家带来了更多的游戏乐趣与策略性挑战。

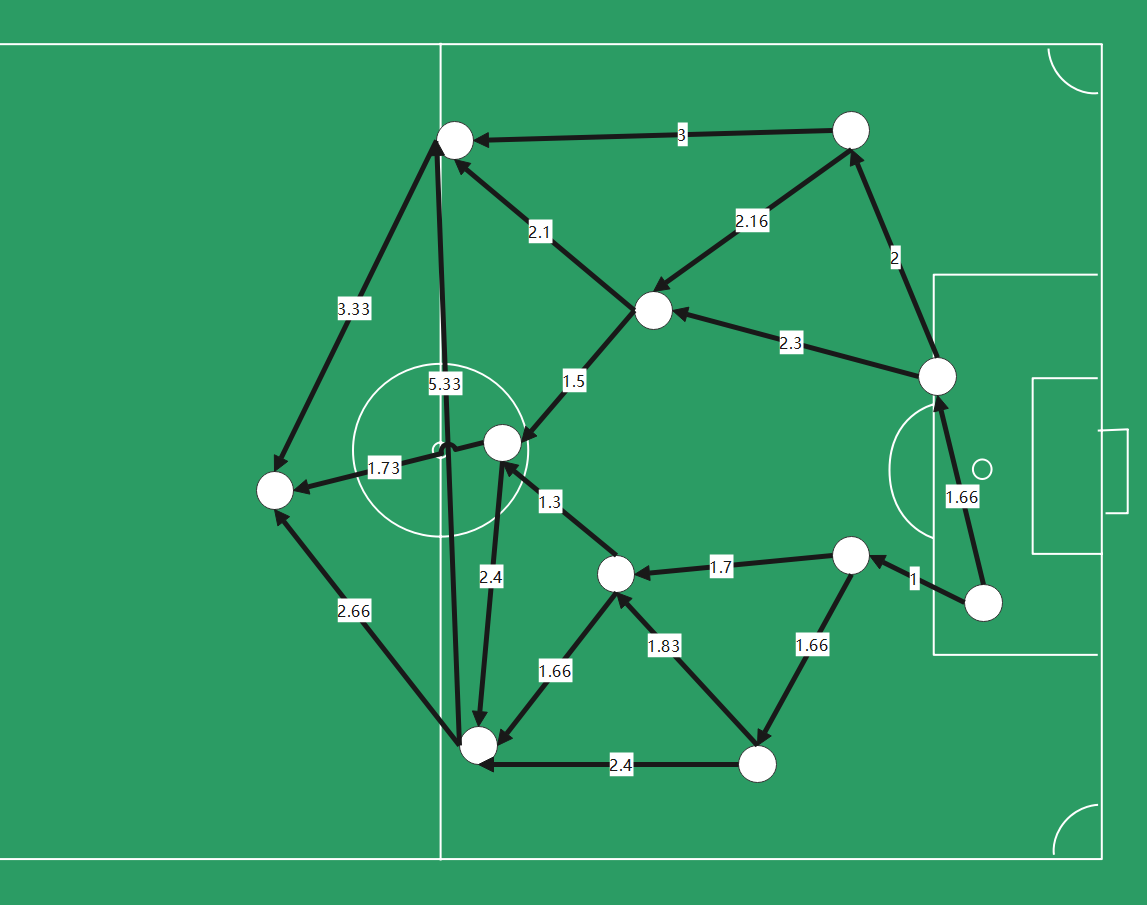


在一场球赛中（如上图所示），敌方攻势被我方拦截后，我方守门员开始准备传球，那么玩家该如何传球才能最快传到队内前锋的脚下呢？（由于本场比赛我方采用4-2-3-1阵型，所以只有一个单前锋）。

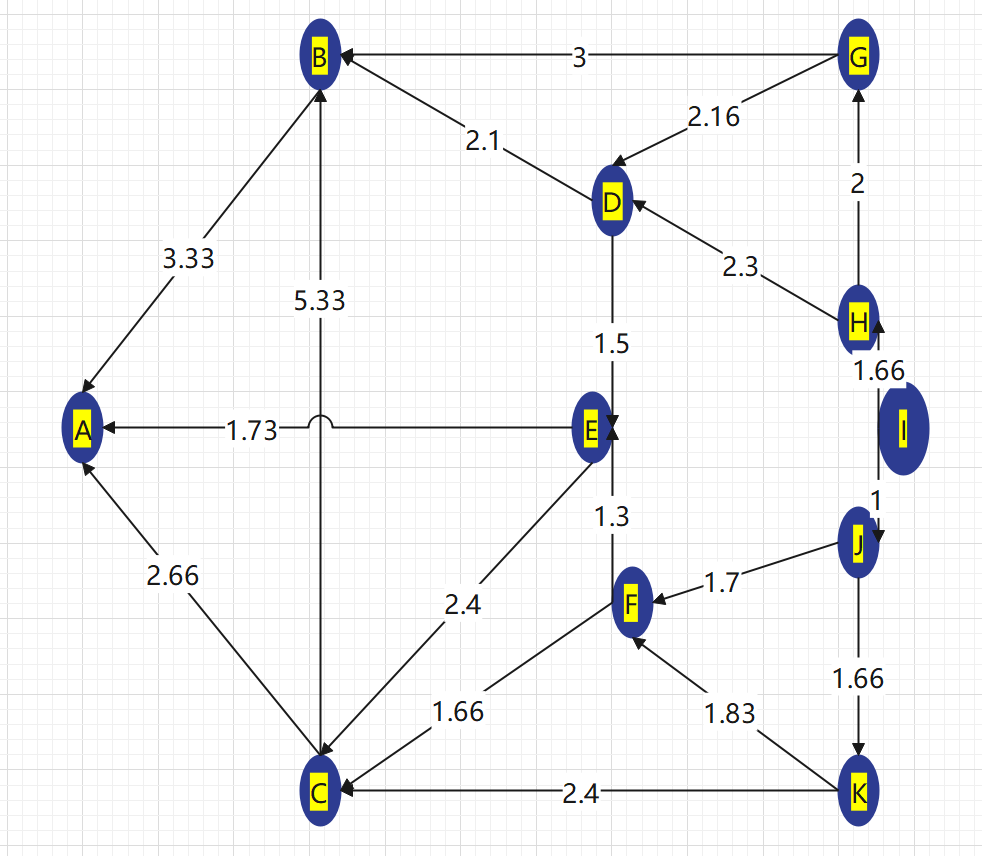
**首先我们需要对我方球员在球场上的位置进行建图G（V，E）。在建图之前，为了简化问题，我们先阐述几条说明：**

1. 该图G为有向加权简单图。
2. 图上的每个点代表对应位置的球员。
3. 每条有向边表示起点位置球员传球给终点位置球员。
4. 由于长传容易丢失球权，所以我们只使用贴地传球。
5. 每个球员的传球只能向前传，不允许往回传。
6. 每个球员的传球选择只能向据他最近的两个己方球员传球。

最终，在球场上的建图如下图所示：



对该图进行简单的调整，得到清楚明了的下图：



**为求解该加权有向简单图的最短路径问题，本篇文章采用Dijkstra算法。**

Dijkstra算法是经典的最短路径求解算法，能求解一个顶点到所有顶点的最短路径。

**大致解释如下：**

1. 设置两点顶点的集合S 和V，集合S 中存放已找到最短路径的顶点，集合V中存放当前还未找到的最短路径的顶点。
2. 初始状态时，集合S中只包含源点，设为V0。
3. 然后从集合V中选择到源点V0 路径长度最短的顶点Vi加入到集合S中。
4. 集合S中每加入一个新的顶点Vi都要修改源点V0 到集合V中剩余顶点的当前最短路径值（如果源点V0无法通过S中的点到达某个点，则源点到这个点的距离为∞）。集合S中各顶点新的当前最短路径长度值，为原来的当前最短路径长度值与从源点过顶点Vi到达该顶点的带权路径长度中的较小者。
5. 回到3，此过程不断重复，直到集合V中的顶点全部加入到集合S中为止。

**求解过程（求解没有采用计算机程序辅助，纯手工求解）：**

　第一次循环

1. ：一开始集合S={I}，
2. : 更新路径。Dist（I，J）=1,Dist（I，H）=1.66,Dist（I，A）=∞,Dist（I，B）=∞,

Dist（I，C）=∞,Dist（I，D）=∞,Dist（I，E）=∞,Dist（I，F）=∞，Dist（I，G）=∞,

Dist（I，K）=∞。

　第二次循环

1. ：离I最近的是J，所以把J加入到S中。集合S={I，J}.
2. ：更新路径。Dist（I，J）=1,Dist（I，H）=1.66,Dist（I，A）=∞,Dist（I，B）=∞,

Dist（I，C）=∞,Dist（I，D）=∞,Dist（I，E）=∞,Dist（I，F）=2.7，Dist（I，G）=∞,

Dist（I，K）=2.66。

…… ……

第九次循环

1. ：离集合S最近的点是G，所以把G加入到S中。集合S={I,J,H,K,F,E,C,A,G}
2. ：I->H->G->B的路径长度 < I->J->F->C->B的路径长度。
3. : 更新路径。Dist（I，J）=1,Dist（I，H）=1.66,Dist（I，A）=5.73,Dist（I，B）=6.66,

Dist（I，C）=4.36,Dist（I，D）=3.96,Dist（I，E）=4,Dist（I，F）=2.7，Dist（I，G）=3.66,

Dist（I，K）=2.66。

…… ……

第十一次循环结束。

**上面的求解过程只用文字展示了部分求解过程。**

**每一次循环源点I到各点的最短距离更新过程和路径记录以及每次循环集合S的变化都在下面两图中：**

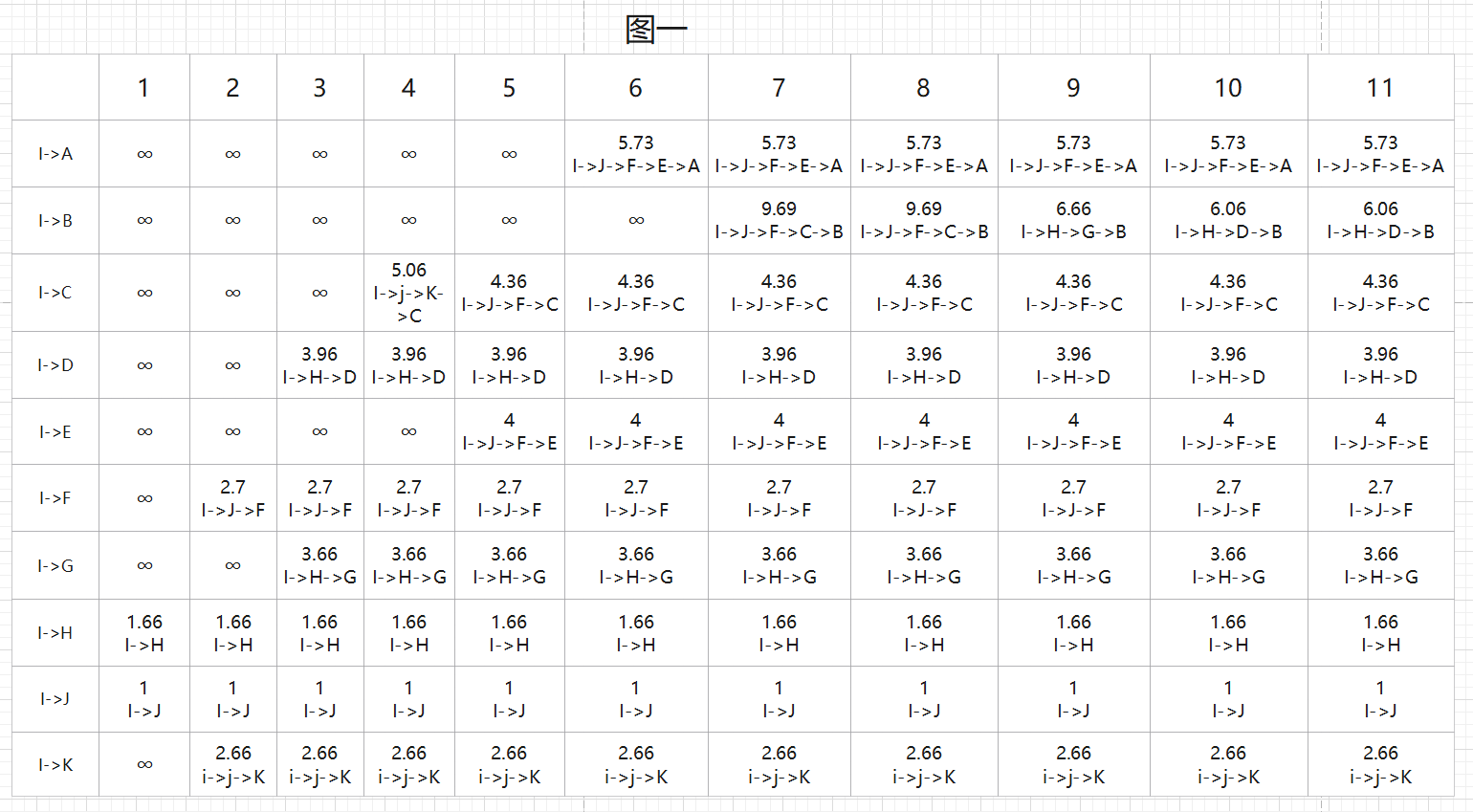
图一表中第一行代表的是第几次循环该算法。

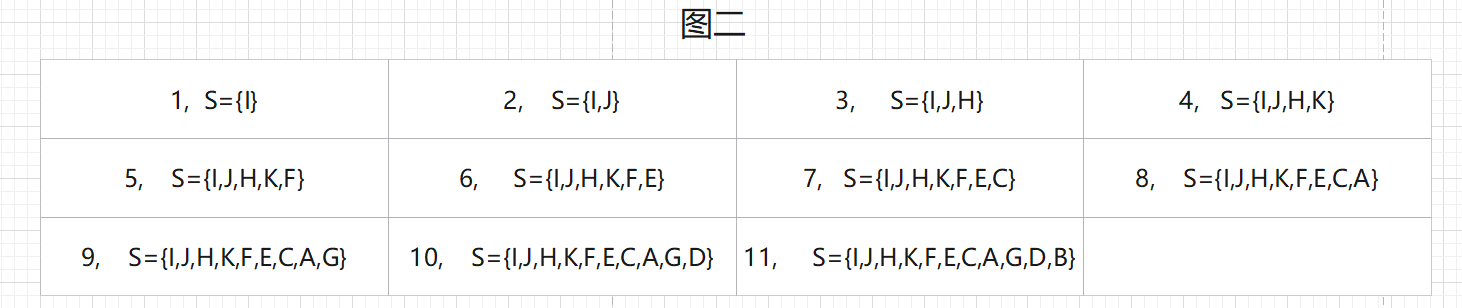
图一表中的第一列表示算法计算出的I到某点的最短距离。

图一表中的其他格子∞代表I还没有通过集合S到该点的路径，其他格子中的第一行表示最短路径的值，第二行表示该最短路径经过的点的顺序。

图一中的最后一次循环也就是最后一列，每一个格子中其实就记录了点I到每个点的最短路径。

图二表中代表每次循环中集合S中的点。





经过求解我们可以得出从守门员I到单前锋A的最短路径为I->J->F->E->A，问题解决。

当然，这样的模型对于实际问题还是过于简单，仍有可以更复杂的情况，如传球不经要考虑己方球员的位置，还要考虑地方球员的威胁。再如，球场上任何球员的位置和距离都是实时变化的，不可能一成不变。希望未来我能有解决更复杂问题的能力。