**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 1**

*дисциплина: Интеллектуальные системы*

Студент: Койфман К.Д.

Группа: НПИбд-01-21

№ ст. билета: 1032217058

**МОСКВА**

2022 г.

**Введение.**

**Цель работы.**

Изучение работы и реализация алгоритмов “Поиска в ширину”,” A\*” и “Дейкстры”.

**Задачи.**

1. Реализовать алгоритмы - A\*, Дейкстры.
2. Реализовать поддержку различных эвристических функций - Euclid, Manhattan, Octile.
3. Реализовать поддержку коэффициента эвристики (f = g+w\*h)
4. Реализовать поддержку 4- и 8-связных графов. (При проверке диагональных переходов необходимо учитывать проходимость смежных вершин).
5. Протестировать реализацию на всех заданиях, используя различные комбинации входных параметров:

5.1. BFS (другие параметры игнорируются)

5.2. Dijkstra (другие параметры игнорируются)

5.3. AStar, connections=4, metrictype=Euclid, hweight=1

5.4. AStar, connections=4, metrictype=Manhattan, hweight=1

5.5. AStar, connections=8, metrictype=Euclid, hweight=1

5.6. AStar, connections=8, metrictype=Octile, hweight=1

5.7. AStar, connections=8, metrictype=Manhattan, hweight=1

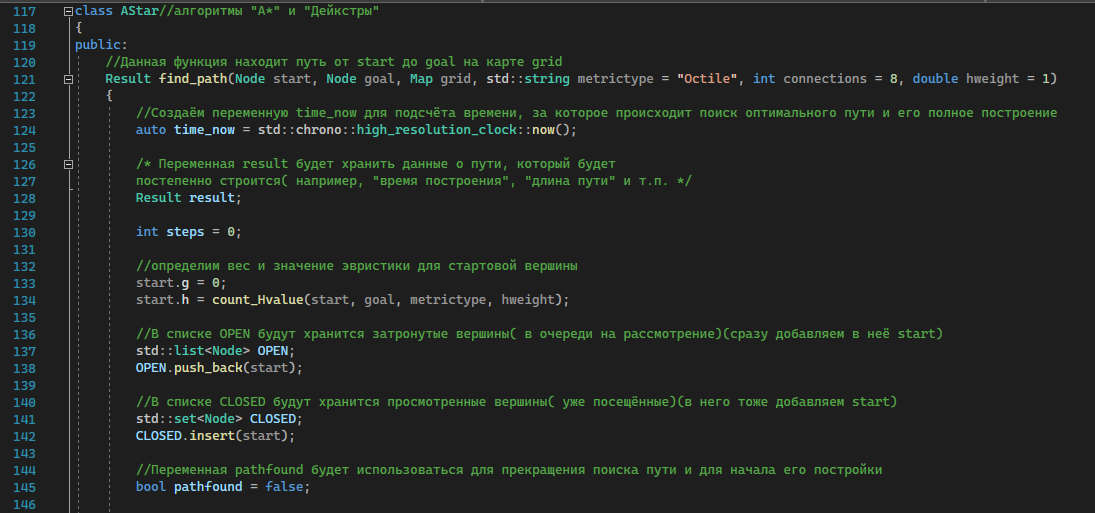
5.8. AStar, connections=8, metrictype=Octile, hweight=2

**Ход работы.**

**1 задание.**

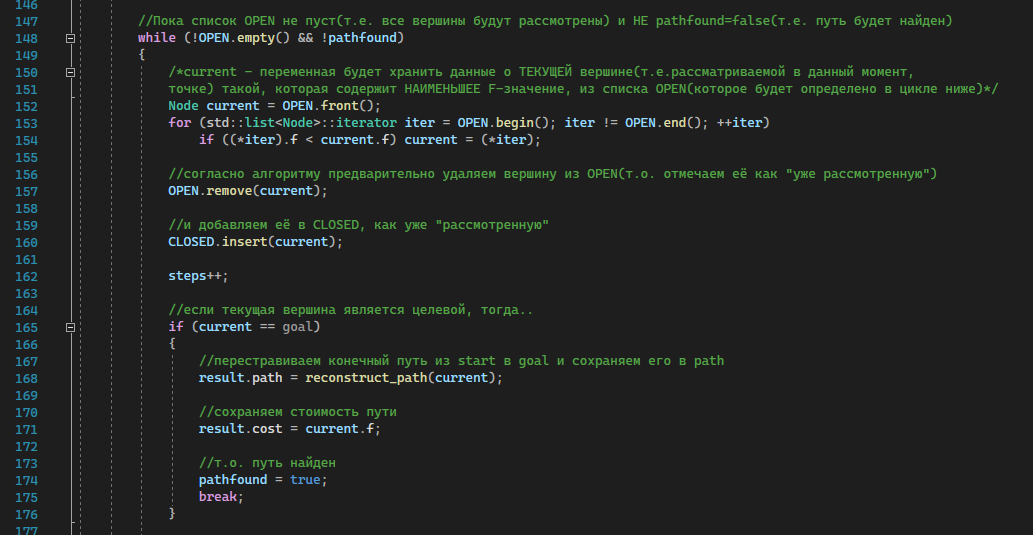
Реализуем алгоритм Дейкстры и А\*.

Определяем начальные условия работы алгоритма и формируем ключевые списки “OPEN” и “CLOSED” (рис.1).



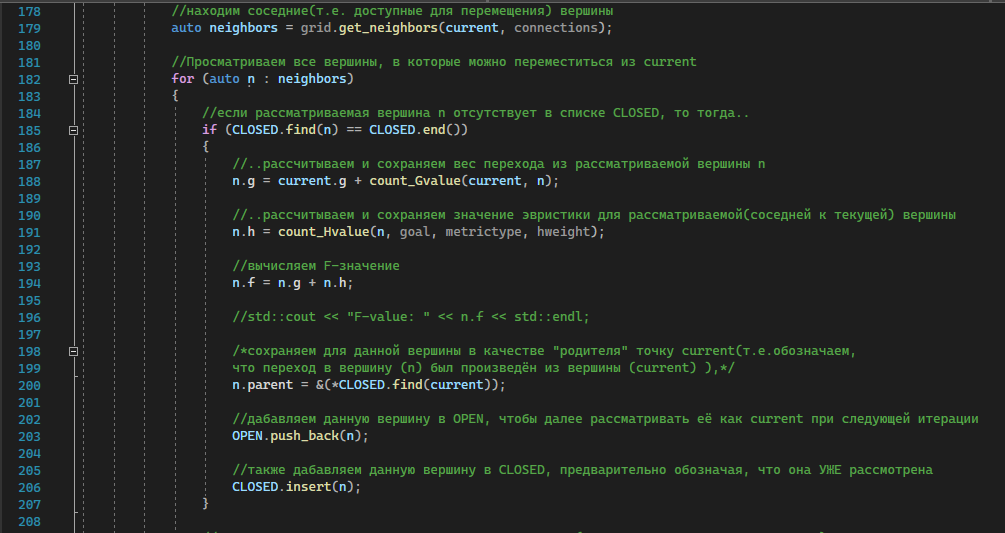
**рис.1**

Открываем цикл while, в течении работы которого алгоритм будет просматривать и раскрывать выгодные для перехода вершины(рис.2):

****

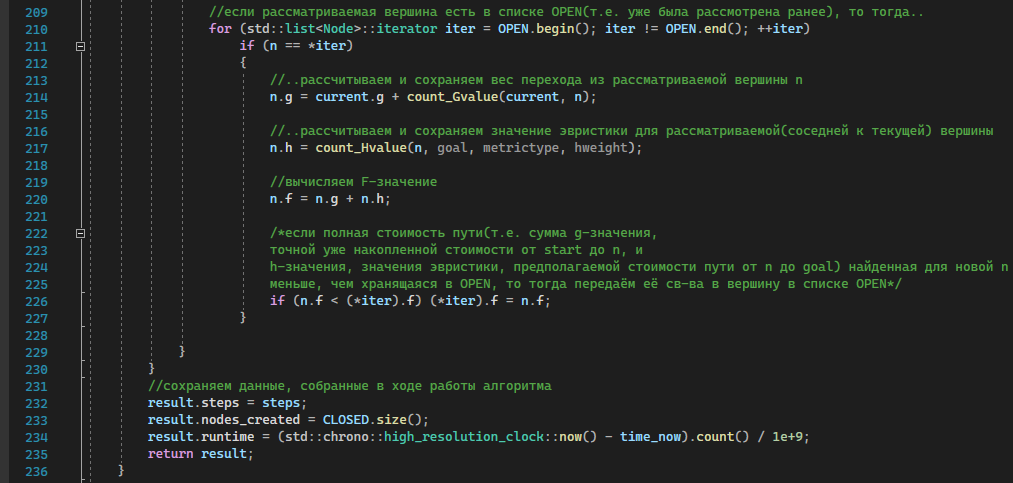
**рис.2**

Найдём вершины, соседние для текущей, доступные для перехода, и проверим не были ли они образованы и/или раскрыты ранее(рис.3):



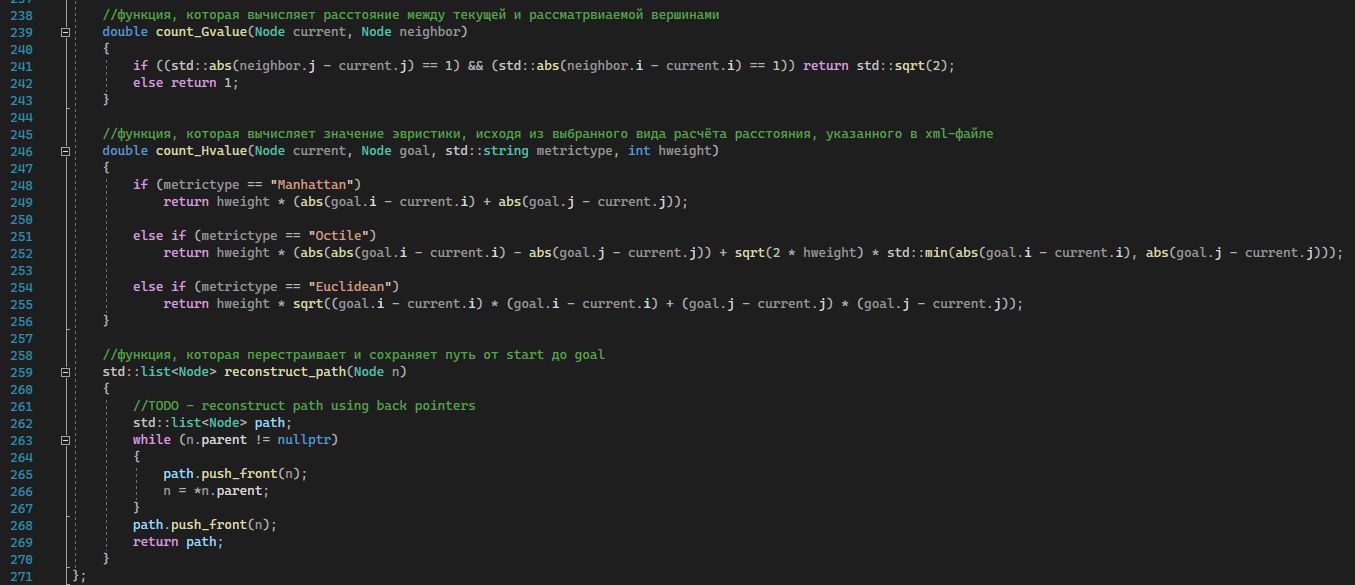
**рис.3**

Проверим, является ли рассматриваемая вершина дубликатом (т.к. она уже могла быть рассмотрена и наделена f-значением). Если – да, то тогда необходимо проверить является ли новое f-значение меньше того, которое было сохранено ранее(рис.4):

****

**рис.4**

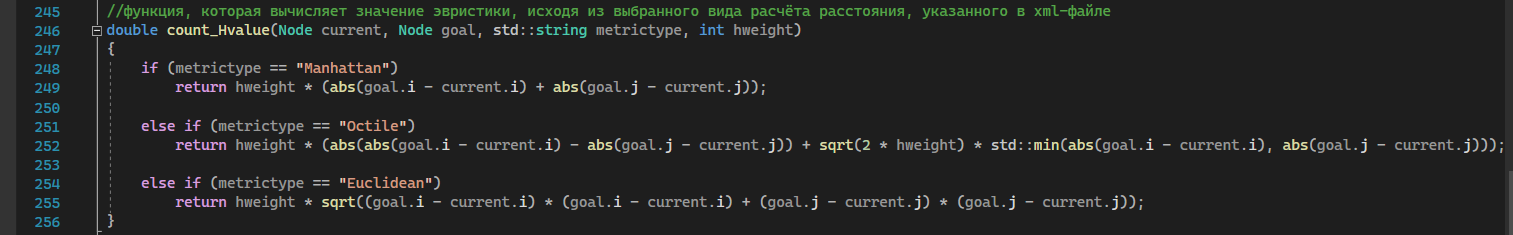
Функции, применённые при работе алгоритма(рис.5):



**рис.5**

**2,3 задание.**

Реализуем поддержку эвристических функций - Euclid, Manhattan, Octile и поддержку коэффициента эвристики (f = g+w\*h) (рис.6):



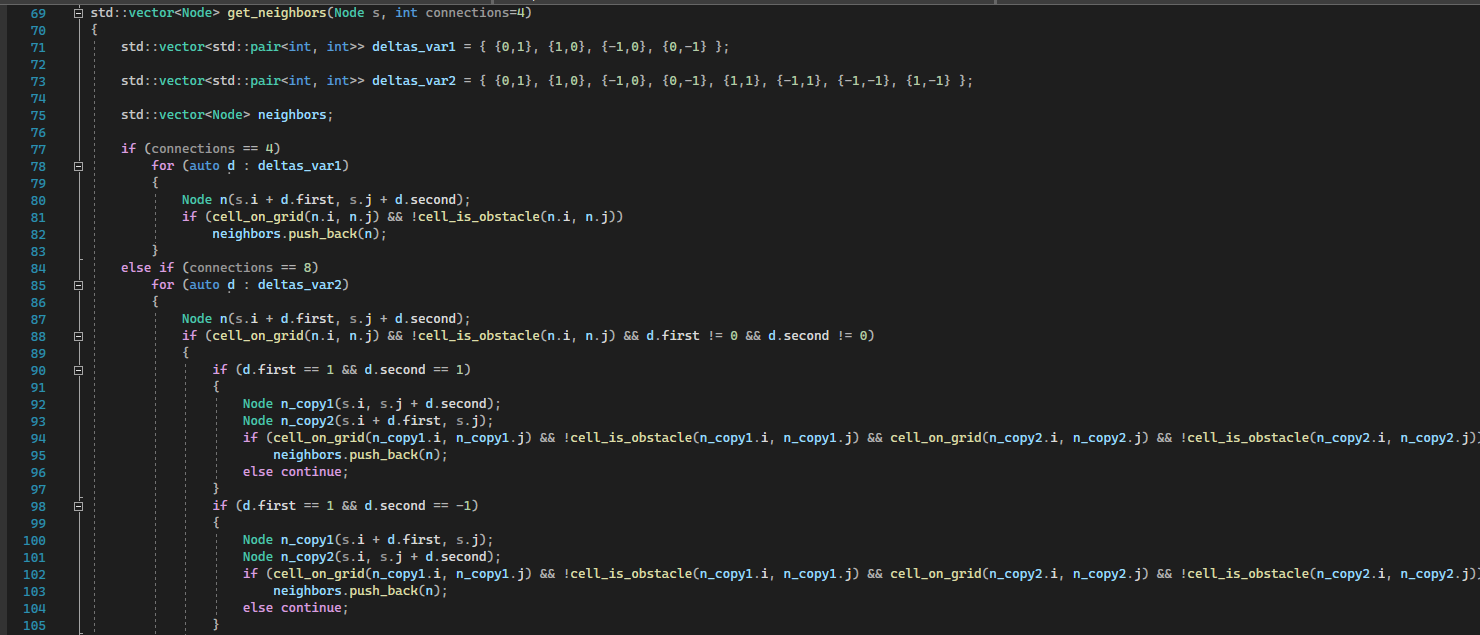
**рис.6**

**4 задание.**

Реализуем поддержку 4- и 8-связных графов так, чтобы при проверке диагональных переходов учитывалась проходимость смежных вершин (рис.7, рис.8):

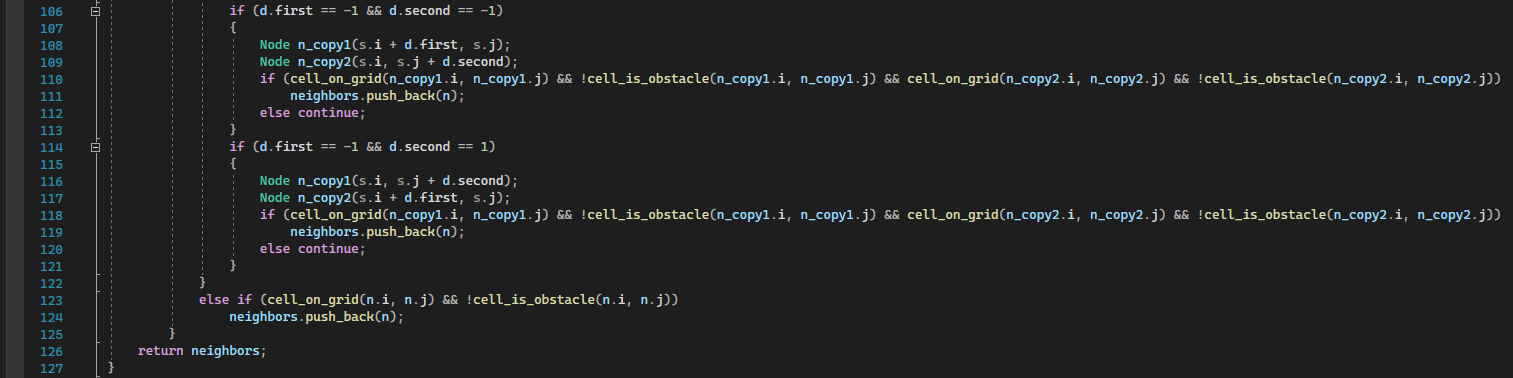
Для этого зададим 2 вектора: 1-ый – для переходов в 4 направлениях, и 2-ой – для переходов в 8 направлениях (т.е. также и по диагонали). Определять, какая вершина является диагональной, мы будем, утверждая, что мы изначально находимся в точке с координатами (0,0) и диагональная вершина будет обладать не нулевыми значениями обеих координат (т.е. x! =0 &&

y! =0):



**рис.7**

Далее в зависимости от координат диагональной вершины будем делать следующее. Для проверки того, можно ли совершить переход по диагонали будет служить условие, согласно которому, если 2 соседние вершины, находящиеся по направлению к вершине, в которую планируется совершить переход, являются клетками карты (т.е. находятся в её пределах) и не являются препятствиями, то переход в указанную вершину может быть произведён(рис.8):



**рис.8**

**5 задание.**

Протестируем реализацию на всех заданиях, используя различные комбинации входных параметров:

**5.1.** BFS (другие параметры игнорируются):

**5.2.** Dijkstra (другие параметры игнорируются):

**5.3.** AStar, connections=4, metrictype=Euclid, hweight=1

**5.4.** AStar, connections=4, metrictype=Manhattan, hweight=1

**5.5.** AStar, connections=8, metrictype=Euclid, hweight=1

**5.6.** AStar, connections=8, metrictype=Octile, hweight=1

**5.7.** AStar, connections=8, metrictype=Manhattan, hweight=1

**5.8.** AStar, connections=8, metrictype=Octile, hweight=2

Исходя из полученных результатов, можно сделать следующие выводы:

* Алгоритм поиска кратчайшего пути А\* является наиболее эффективным среди остальных, так как срабатывает за минимальное число итераций
* За счёт использования формулы расчёта диагонального расстояния (“Octile metric type”) в связке с алгоритмом A\* значительно сокращается время, число раскрытых вершин и число итераций работы алгоритма
* Алгоритм BFS тратит меньше всех остальных времени на поиск оптимального пути
* Алгоритм Dijkstra чаще остальных находит самый коротки путь

**Заключение.**

В ходе проделанной лабораторной работы, мной были усвоены основные принципы работы алгоритмов “A\*”, “Дейкстры” и “Поиска в ширину”.