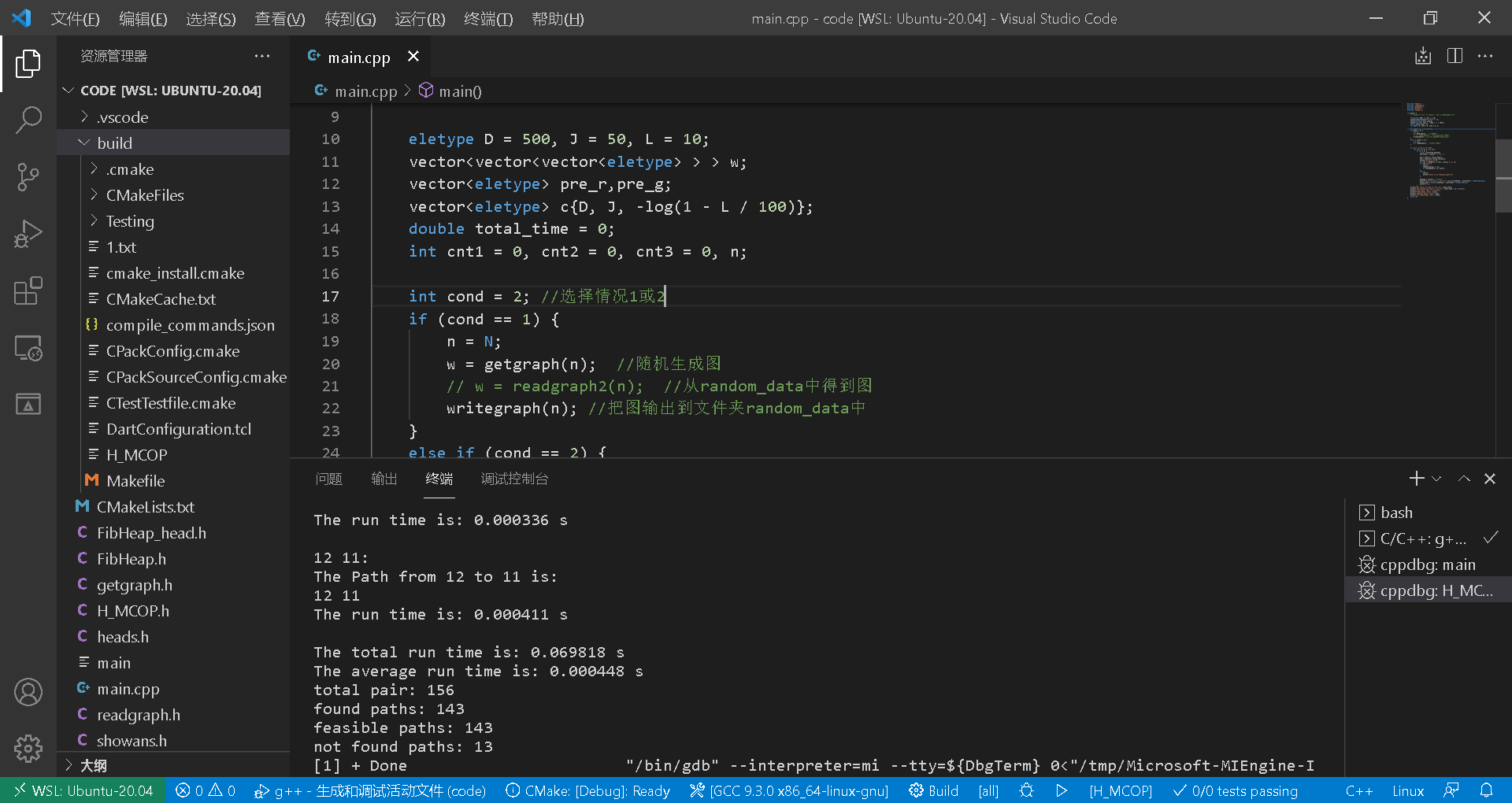
# H\_MCOP算法实验报告

无98 杨元昊 2019011197

1. “算法运行示例”截图

* 情况一：图100个节点。三种属性随机生成。
* 输出包括：每对节点之间选出的路径经过的节点，每对节点之间计算过程的计算时间，以及总计算时间和平均计算时间。同时把随机生成的100个节点的三个属性输出到文件夹random\_data里。
* 情况二：图13个节点。三种属性从data文件夹中读取。
* 输出包括：每对节点之间选出的路径经过的节点，每对节点之间计算过程的计算时间，以及总计算时间和平均计算时间。
* 下面的截图为情况二的运行结果截图。（情况一只需把int cond = 2;改为int cond = 1;）
* 

1. 变量说明

* 函数参数：
* 在函数H\_MCOP(w, c, pre\_r, pre\_g, n, s, t)中，w表示链路的加性权重，c表示权重的约束，pre\_r表示Reverse\_Dijkstra函数计算过程中节点u的前驱结点，pre\_g表示Look\_Ahead\_Dijkstra函数计算过程中节点u的前驱结点，n表示节点的个数，s表示源点，t表示汇点。
* 在函数reserve\_dij(w, c, pre\_r, n, t)中，各个参数的意义同上。
* 在函数look\_ahead\_dij(w, c, pre\_g, n, s)中，各个参数的意义同上。
* 在函数Prefer\_the\_best(v, Ctmp, gtmp, Gtmp, Rtmp, c)中，v表示松弛操作(u, v)中的v，Ctmp，gtmp，Gtmp，Rtmp表示临时节点tmp的各个标签值，c表示权重的约束。
* 全局变量：
* 在H\_MCOP.h中，对应Reverse\_Dijkstra部分的：r表示当代价函数为g1时节点u到t的代价；R表示每部分链路权重的累积量。
* 对应Look\_Ahead\_Dijkstra部分的：g表示当代价函数为h时，经过节点u的从s到t的路径的代价；C表示节点s到u的主权重；G表示每种链路权重的累积量。
* 在getgraph.h中，lat，jit，plr分别表示往返时延，时延抖动，丢包率。
* 在showans.h中，ans表示输出的路径。
* 在heads.h中，N表示情况一的节点数100，Dim表示情况二的节点数13，K表示约束的个数3，Inf表示设定一个足够大的数1e6即。

1. 算法运行时间的统计结果

* 规定算法中主权重为往返时延。
* 设定时延、抖动和丢包率的上限分别为500 ms，50 ms，10 %。
* 情况一中随机的数据精度为小数点后两位，时延在0 - 1000 ms间均匀随机，抖动在0 - 100 ms间均匀随机，丢包率在0 - 10 %间均匀随机。
* 情况一：一共计算100 \* 99 个有序节点对，总计算时间为266.749497 s，平均每对计算时间为0.026944 s。
* 情况二：一共计算13 \* 12个有序节点对，总计算时间为0.056231 s，平均每对计算时间为0.000360 s。
* 完整输出结果保存在"情况1输出结果.txt"和"情况2输出结果.txt"中。