

EE6094
CAD for VLSI Design
Programming Assignment 4 Report

Student Name:顏郁芩

Student ID:106501001

一、題目描述

利用 C++ 寫一個程式可以把題目給的 Block 排出一個最小的面積，輸出排出來的邊長、總面積還有個別方塊的左下角座標和邊長。

要注意每一個方塊的邊長比例要在限制範圍內、每一個方塊不能重疊還有面積誤差不能大於 1%

二、實現過程

程式架構分成讀資料還有做交換的兩個大部分，第一個讀資料的部分就是差不多跟之前的 PA 相同都是把資料讀進來做處理，第二個部分就是主要做交換的地方，那個部分是參考老師上課講解過的虛擬碼架構來寫的。

資料存取跟上次一樣都是採用 **struct 搭配 vector** 的方式存的，分成 **block 區塊**(存輸入已知的訊息還有增加的訊息)，還有動態存取的 **leaf vector**。E、NE 和 **best** 也是用 **vector** 來儲存。

輸入資料存進 **struct** 裡面之後，為了**創造初始的 E** 想了兩個方法：

方法 1：BLOCK+運算子+BLOCK+運算子……………

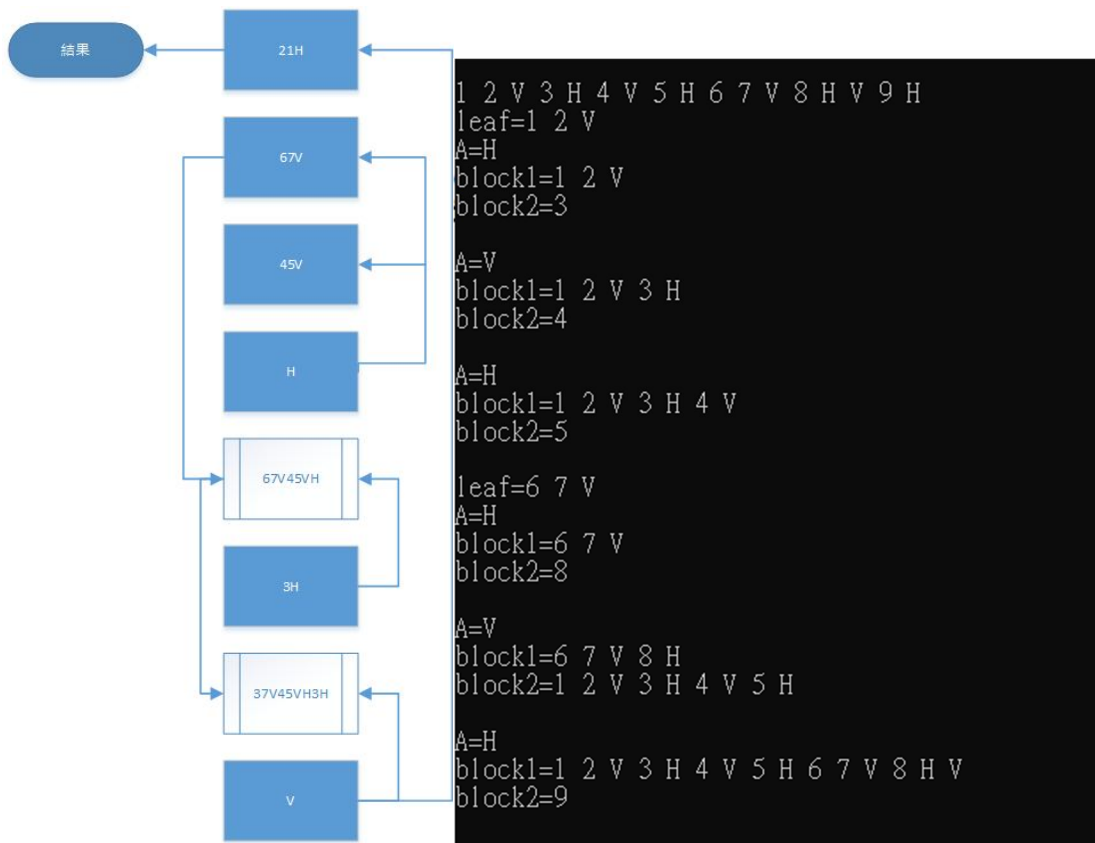
方法 2：BLOCK +BLOCK+運算子+ BLOCK +BLOCK+運算子+運算子+ BLOCK+運算子+BLOCK+運算子……………

程式裡面比重占比較多的部分，可以分成計算總面積(**cost**)還有左下角座標兩個：

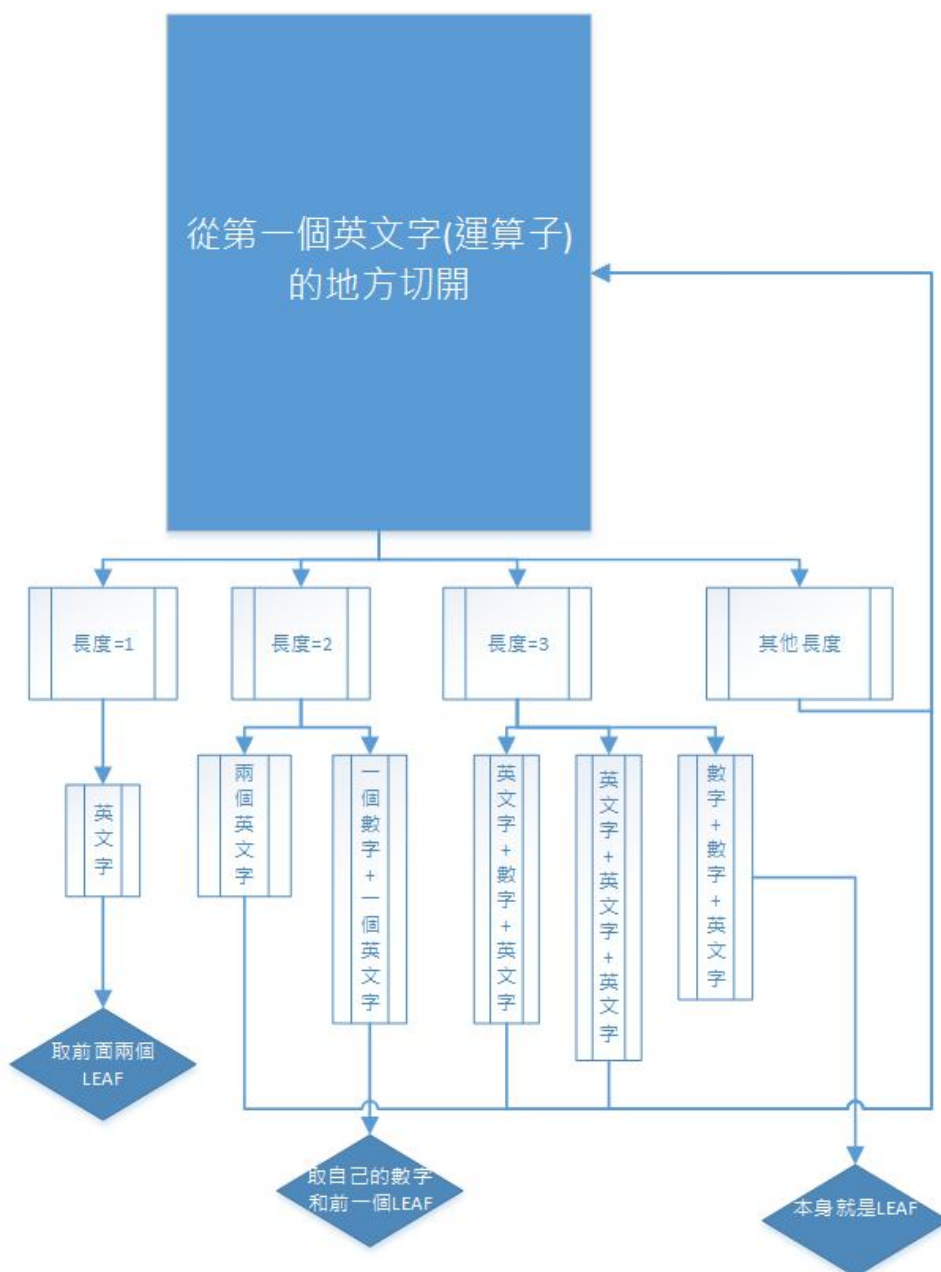
計算總面積(包含 block 長寬)：

首先是先分析一下組成，以 21H67V45VH3HV 為例子，可以看到以下的狀況

1. 如果是讀到 21H、67V、45V 之類的，就可以直接成 LEAF
2. 要是讀到 H 的話，就要把前面兩個 LEAF 加起來存成新的 LEAF
3. 讀到 3H 的話就是自己的 3 加上前一個 LEAF 加起來儲存。

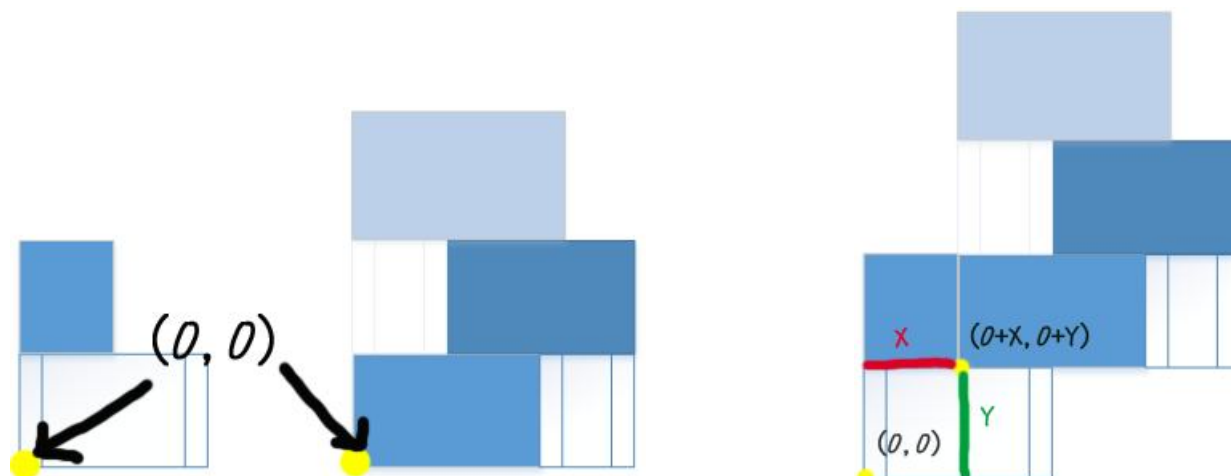


從以上的結果衍伸出以下的程式碼，把排出來的向量丟進去分析，可以同時計算疊起來的 LEAF 長寬、個別的長寬還有左下角的座標。



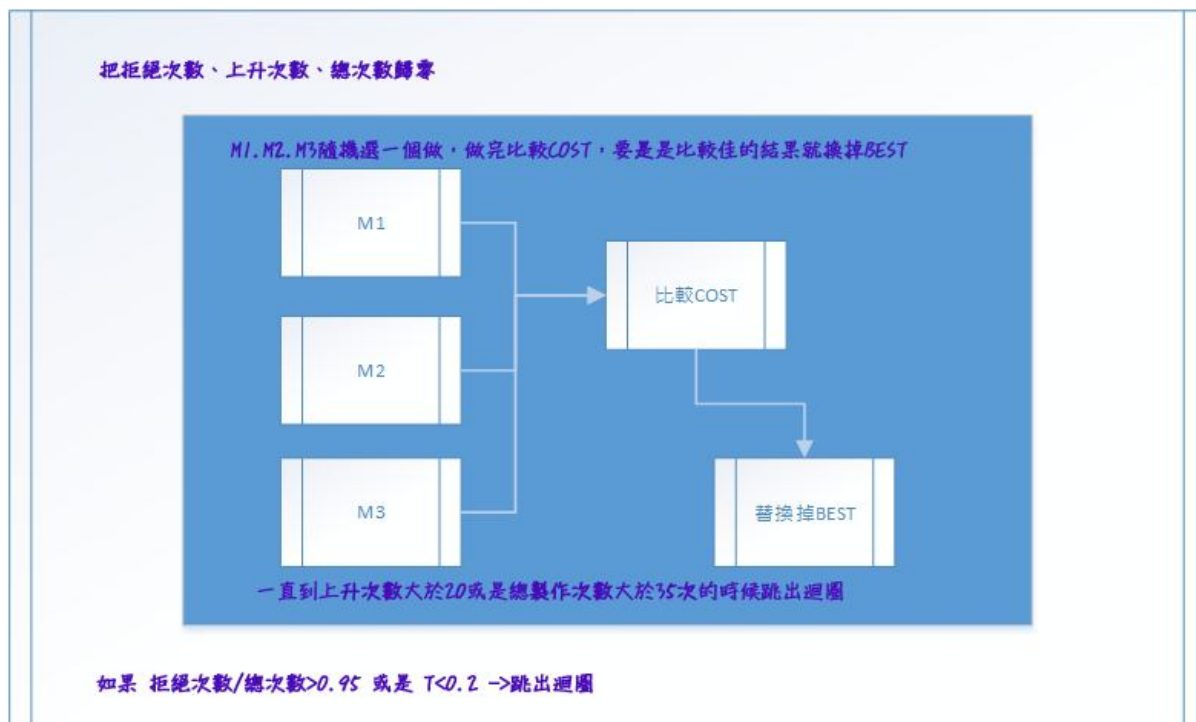
計算左下角座標的概念：

假設有兩個不同的 LEAF，兩個的左下角都是(0,0)，然後要疊成右圖那張的話，另一個 LEAF 的左下角座標就會要加上(X,Y)，同樣的，那個 LEAF 裡面的所有 block 的座標也要跟著變動(X,Y)



三、程式碼

交換的程式碼部分架構(利用上課講解的虛擬碼寫成的)：



用變數讓三個交換方式隨機執行，M1：交換 BLOCK，M2：把 V 換 H，H 換 V，M3 的話是把 BLOCK 和運算子交換，但是要滿足在運算子的數量不能大於運算元數量。

自己寫完的程式碼裡面目前有發現的 BUG 以及可能的解決方法

問題	解決方法/可能造成此問題的原因
算出來的左下角座標會導致重疊	可能在計算座標過程中有什麼地方沒有考慮到
會維持最原本的相似結構不變(只有 M1，M2 有效)	可能因為初始的 VECTOR 創造方式就讓 M3 無法執行，可以試著改變初始的創造模式，或是像上課提到的，在某個時間之前做某一種變化，強制將整個結構改掉
可能沒有辦法找到最佳解	時間限制有寫進去但並沒有用上，而是限制了交換次數，可能因為交換次數不夠多所以沒有辦法找到最佳解

四、測資結果

T200 和 T300 兩筆測資的個別 BLOCK 資料太多不知道怎麼截圖所以把輸出檔案直接付在壓縮檔裡面，其他都是工作站上面的結果。

t10.txt

```
[106501001@eda359_forclass ~/PA4]$ ./go t10.txt t10_out.txt
0 1 V 2 3 H V 4 V 5 H 6 H 7 H 8 H 9 H

58.21 438.45 25522.2
0 301.76 4.47 3.58
0 0 37.2 63.23
88.54 98.95 54.41 92.49
88.54 0 58.21 98.95
0 191.44 46.48 37.18
0 271.4 37.95 30.36
0 360.73 58.21 77.72
37.2 0 51.34 87.27
0 228.62 53.48 42.78
0 305.34 58.21 55.39
1 7 V 3 2 H V 4 H 8 H 5 H 0 H 9 H 6 H
time consume: 0.02
```

t20.txt

```
[106501001@eda359_forclass ~/PA4]$ ./go t20.txt t20_out.txt
0 1 V 2 3 H V 4 V 5 H 6 H 7 H 8 H 9 H 10 H 11 H 12 H 13 H 14 H 15 H 16 H 17 H 18
H 19 H

353.57 5779.58 2.04349e+06
0 1017.58 309.59 526.3
0 574.96 369.67 628.45
0 5555.48 280.12 224.1
0 1434.3 59.76 101.6
0 3569.24 183.64 146.91
0 2186.28 353.57 601.07
0 2787.35 353.57 535.11
0 0 338.21 574.96
0 5281.38 342.62 274.1
0 1543.88 309.59 517.5
0 2061.38 156.12 124.9
0 4422.89 353.57 396.23
0 1203.41 135.82 230.89
0 542.53 283.18 475.05
0 3716.15 242.98 194.39
0 3910.54 353.57 512.35
0 332.49 262.55 210.04
0 4819.12 353.57 462.26
0 3322.46 308.48 246.78
135.82 0 147.36 184.2
7 1 H 12 3 H H 19 V 16 H 13 H 0 H 9 H 10 H 5 H 6 H 18 H 4 H 14 H 15 H 11 H 17 H
8 H 2 H
time consume: 0.06
```



```
63.1 1678.41 105908  
0 0 41.84 71.13  
0 854.39 63.1 78.61  
108.46 0 43.66 74.22  
108.46 74.22 51.34 87.27  
41.84 0 66.62 113.25  
0 268.75 63.1 94.14  
0 1226.02 63.1 94.14  
0 384.28 63.1 53.31  
0 437.59 61.48 49.19  
0 362.89 26.74 21.39  
0 532.18 42.25 33.8  
0 825.82 35.71 28.57  
0 636.34 63.1 56.48  
0 692.82 63.1 89.26  
0 782.08 54.68 43.74  
0 1129.53 63.1 62.5  
0 161.49 63.1 107.26  
0 933 39.69 31.75  
0 964.75 63.1 76.07  
0 1040.82 47.96 38.37  
0 1079.19 62.93 50.34  
0 486.78 56.75 45.4  
0 1192.03 42.49 33.99  
0 565.98 63.1 70.36  
0 1320.16 63.1 56.99  
0 1377.15 53.62 42.9  
0 1420.05 63.1 96.35  
0 1516.4 21.21 16.97  
0 1533.37 63.1 82.54  
0 1615.91 63.1 62.5  
0 4 V 2 3 H V 16 H 5 H 9 H 7 H 8 H 21 H 10 H 23 H 12 H 13 H 14 H 11 H 1 H 17 H 18 H 19 H 20 H 15 H 22 H 6 H 24 H 25 H 26 H 27 H 28 H 29 H  
time consume: 0.13  
[macF040340e-d-0f2-func] [DA13A]
```

```
107 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
1 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
3 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
5 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123
125 124 10 126 117 128 129 130 131 132 97 134 150 136 109 138 139 140 141 142 50 144 145 146 147 148 149 150
23 151 152 153 154 27 156 157 158 159 192 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176
177 178 179 180 181 113 183 184 185 186 187 188 189 190 191 160 193 194 195 196 197 198 199 H
time consume: 3.94
```

0	1	V	2	3	H	V	4	V	5	H	6	H	7	H	8	H	9	H	10	H	11	H	12	H	13	H	14	H	15	H	16	H	17	H	18	H	19	H	20	H	21	H	22	H	23	H	24	H	25	H	26	H	27	H	28	H	29	H	30	H	31	H	32	H	33	H	34	H	35	H	36	H	37	H	38	H	39	H	40	H	41	H	42	H	43	H	44	H	45	H	46	H	47	H	48	H	49	H	50	H	51	H	52	H	53	H	54	H	55	H	56	H	57	H	58	H	59	H	60	H	61	H	62	H	63	H	64	H	65	H	66	H	67	H	68	H	69	H	70	H	71	H	72	H	73	H	74	H	75	H	76	H	77	H	78	H	79	H	80	H	81	H	82	H	83	H	84	H	85	H	86	H	87	H	88	H	89	H	90	H	91	H	92	H	93	H	94	H	95	H	96	H	97	H	98	H	99	H	100	H	101	H	102	H	103	H	104	H	105	H	106	H	107	H	108	H	109	H	110	H	111	H	112	H	113	H	114	H	115	H	116	H	117	H	118	H	119	H	120	H	121	H	122	H	123	H	124	H	125	H	126	H	127	H	128	H	129	H	130	H	131	H	132	H	133	H	134	H	135	H	136	H	137	H	138	H	139	H	140	H	141	H	142	H	143	H	144	H	145	H	146	H	147	H	148	H	149	H	150	H	151	H	152	H	153	H	154	H	155	H	156	H	157	H	158	H	159	H	160	H	161	H	162	H	163	H	164	H	165	H	166	H	167	H	168	H	169	H	170	H	171	H	172	H	173	H	174	H	175	H	176	H	177	H	178	H	179	H	180	H	181	H	182	H	183	H	184	H	185	H	186	H	187	H	188	H	189	H	190	H	191	H	192	H	193	H	194	H	195	H	196	H	197	H	198	H	199	H	200	H	201	H	202	H	203	H	204	H	205	H	206	H	207	H	208	H	209	H	210	H	211	H	212	H	213	H	214	H	215	H	216	H	217	H	218	H	219	H	220	H	221	H	222	H	223	H	224	H	225	H	226	H	227	H	228	H	229	H	230	H	231	H	232	H	233	H	234	H	235	H	236	H	237	H	238	H	239	H	240	H	241	H	242	H	243	H	244	H	245	H	246	H	247	H	248	H	249	H	250	H	251	H	252	H	253	H	254	H	255	H	256	H	257	H	258	H	259	H	260	H	261	H	262	H	263	H	264	H	265	H	266	H	267	H	268	H	269	H	270	H	271	H	272	H	273	H	274	H	275	H	276	H	277	H	278	H	279	H	280	H	281	H	282	H	283	H	284	H	285	H	286	H	287	H	288	H	289	H	290	H	291	H	292	H	293	H	294	H	295	H	296	H	297	H	298	H	299	H	300	H
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

雖然這次自己寫出來的東西沒有完全達到題目的要求，甚至是連 **checker** 都沒有完全正確，但是有對 **floop planing** 的概念更加熟悉。

也嘗試了一些新東西，像是之前沒有用過的時間控制函數，還有很少用到的隨機變數。

<https://www.itread01.com/content/1548646212.html>

<https://www.itread01.com/content/1550239959.html>

<https://mropengate.blogspot.com/2015/07/cc-vector-stl.html>

<https://openhome.cc/Gossip/CppGossip/vector1.html>

使用 vector

<https://blog.csdn.net/test1280/article/details/65937779>

c++中 vector find 使用

<https://www.itread01.com/content/1546853403.html>

C++如何合并若干个 vector

<https://www.coder.work/article/2738995>

c++ - 在 std::vector 中找不到成员