

Rubiks-Cube Solver Parallelizing

0716206 陳昱丞

0716221 余忠旻

309553040 葉柏廷

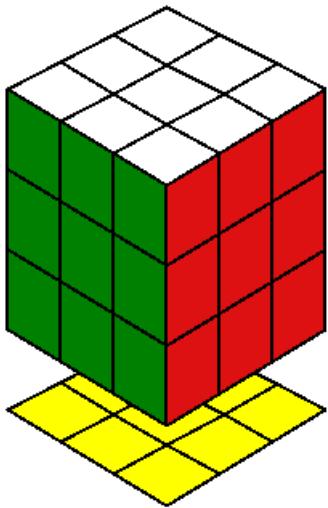
介紹

- ▶ 公式解
- ▶ 上帝的數字：所有三階魔術方塊都可以在有限步數 (n 步) 內還原
- ▶ 2007年：20台超級電腦8000小時證明 $n \leq 26$
- ▶ 2010年：證明 $n \leq 20$ ，現在上帝的數字正式定為20

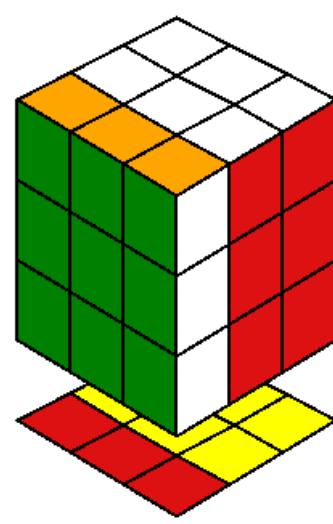
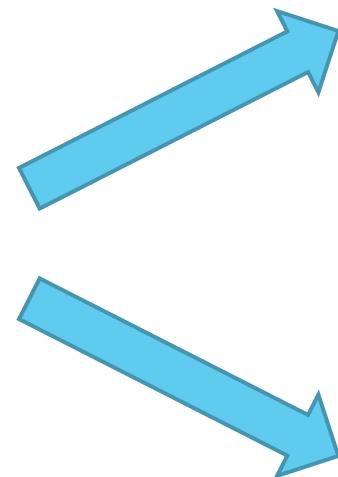
搜尋 - 旋轉方向

- ▶ 六個面，對每個面做順時針及逆時針的旋轉，總共**12種**旋轉方式

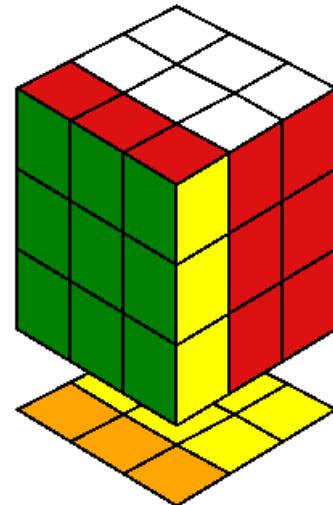
Front



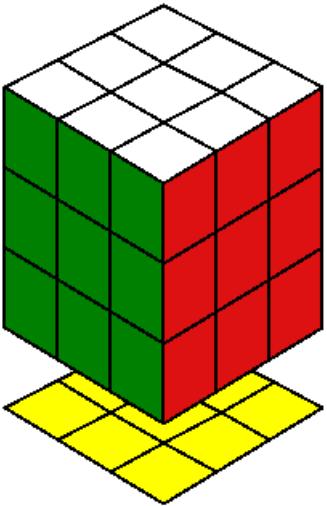
rotate 1:
Front-Clockwise



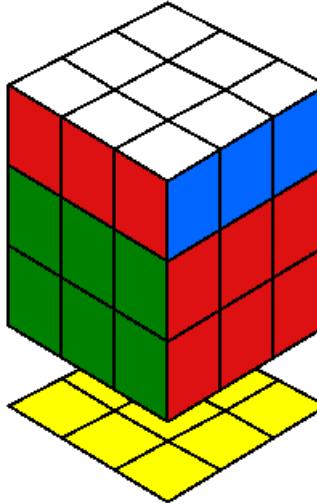
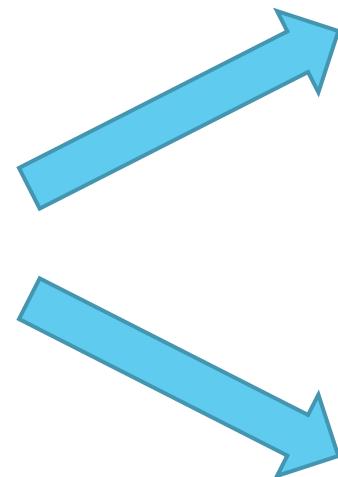
rotate 2:
Front-Anticlockwise



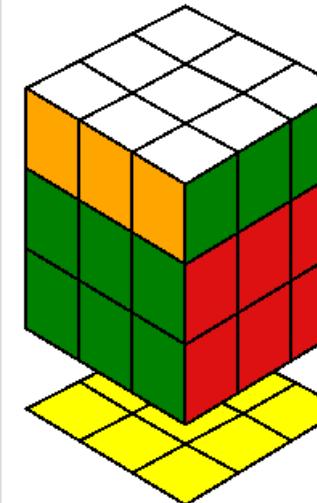
Up



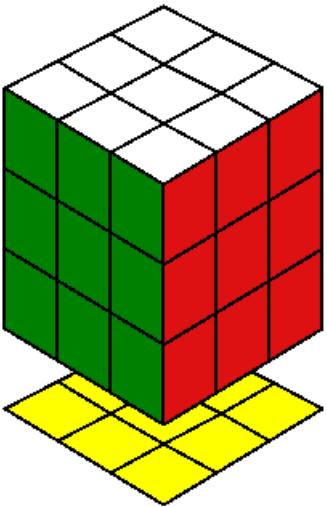
rotate 3:
Up-Clockwise



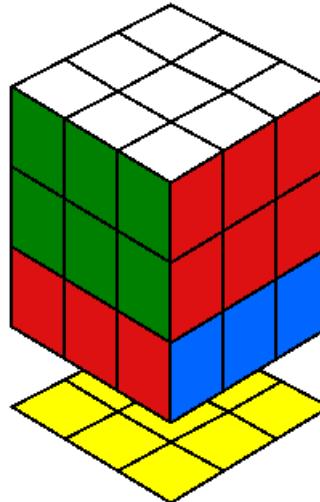
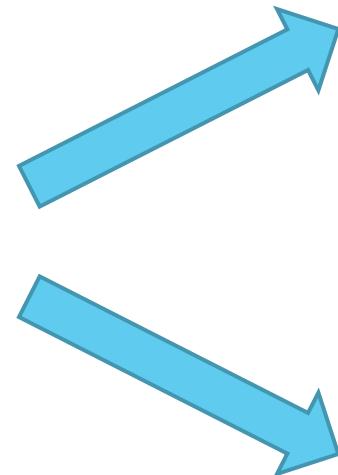
rotate 4:
Up-Anticlockwise



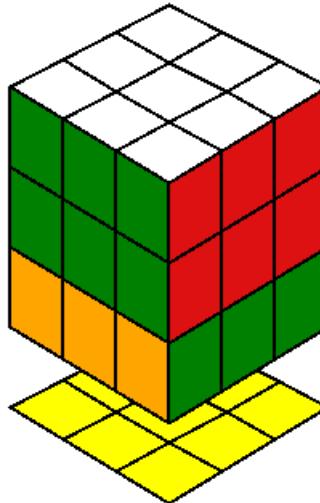
Down



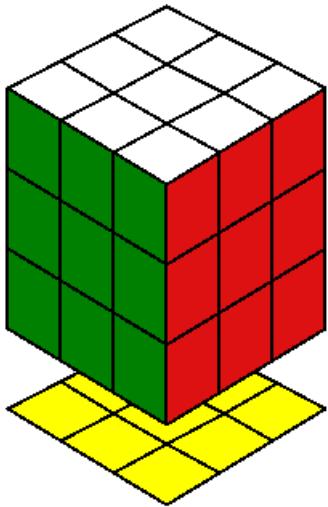
rotate 5:
Down-Clockwise



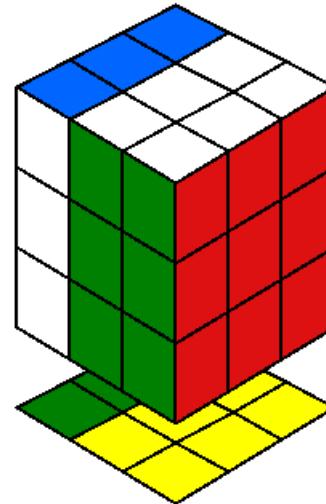
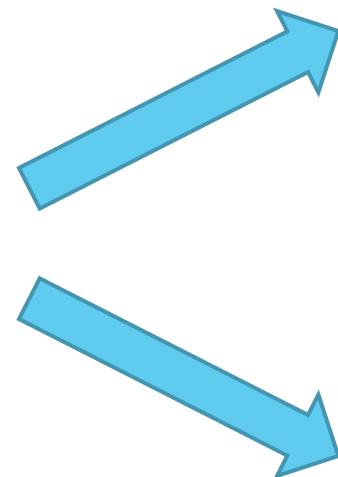
rotate 6:
Down-Anticlockwise



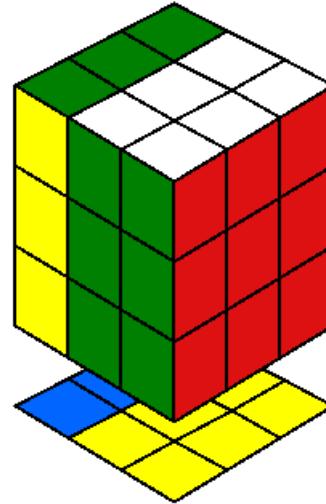
Left



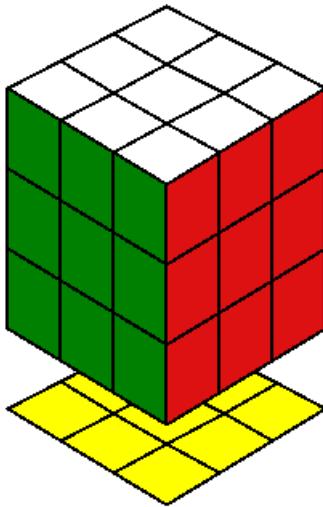
rotate 7:
Left-Clockwise



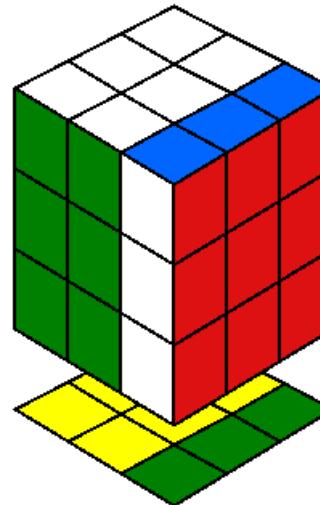
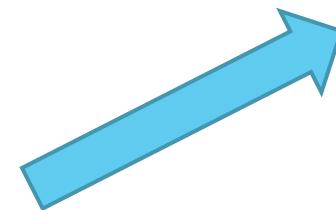
rotate 8:
Left-Anticlockwise



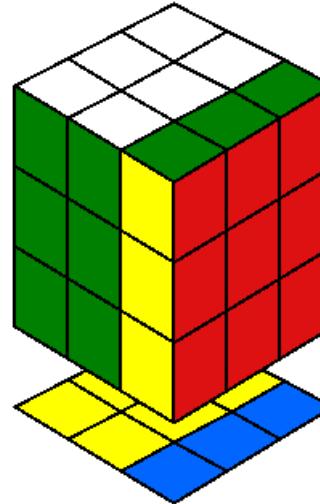
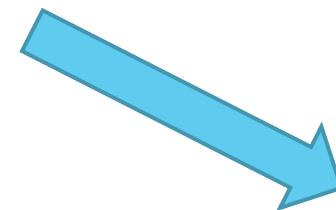
Right



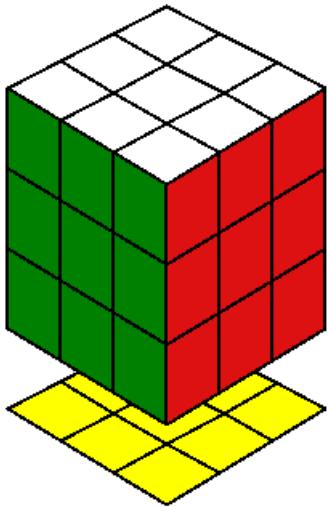
rotate 9:
Right-Clockwise



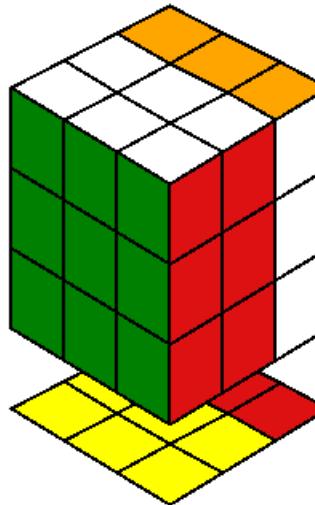
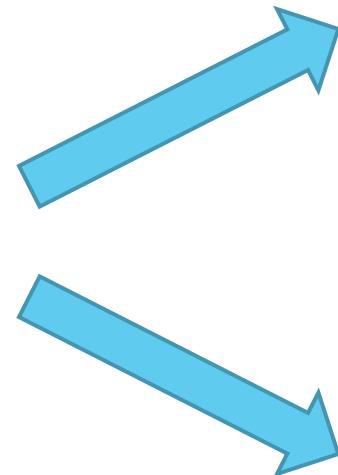
rotate 10:
Right-Anticlockwise



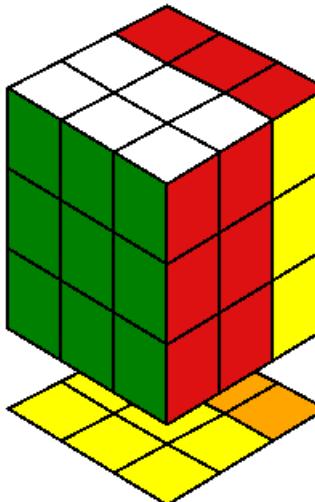
Back



rotate 11:
Back-Clockwise



rotate 12:
Back-Anticlockwise



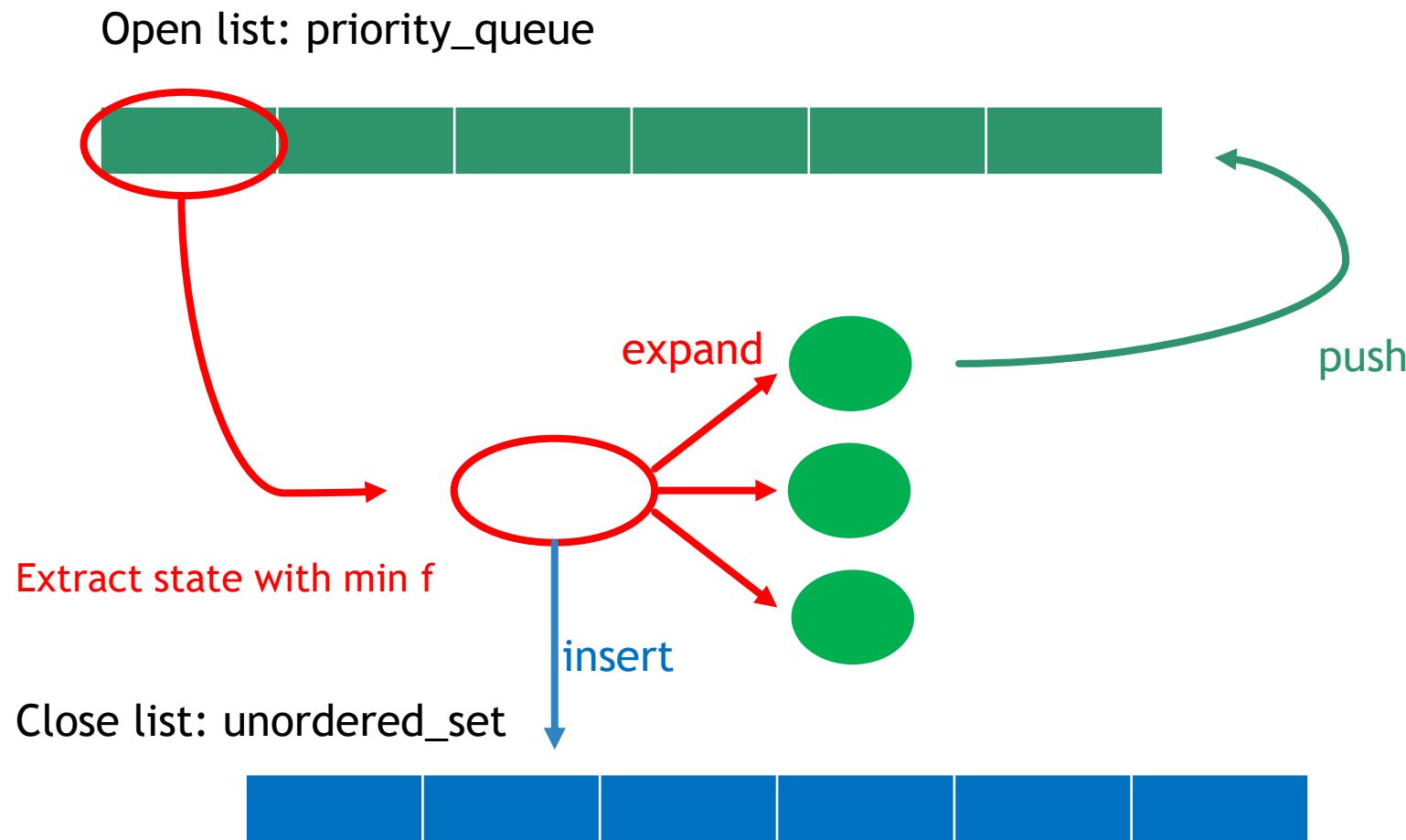
搜尋架構 - Serial

- ▶ Breadth First Search (BFS)
- ▶ A star (A^*)
- ▶ Iterative Deepening A star (IDA^*)

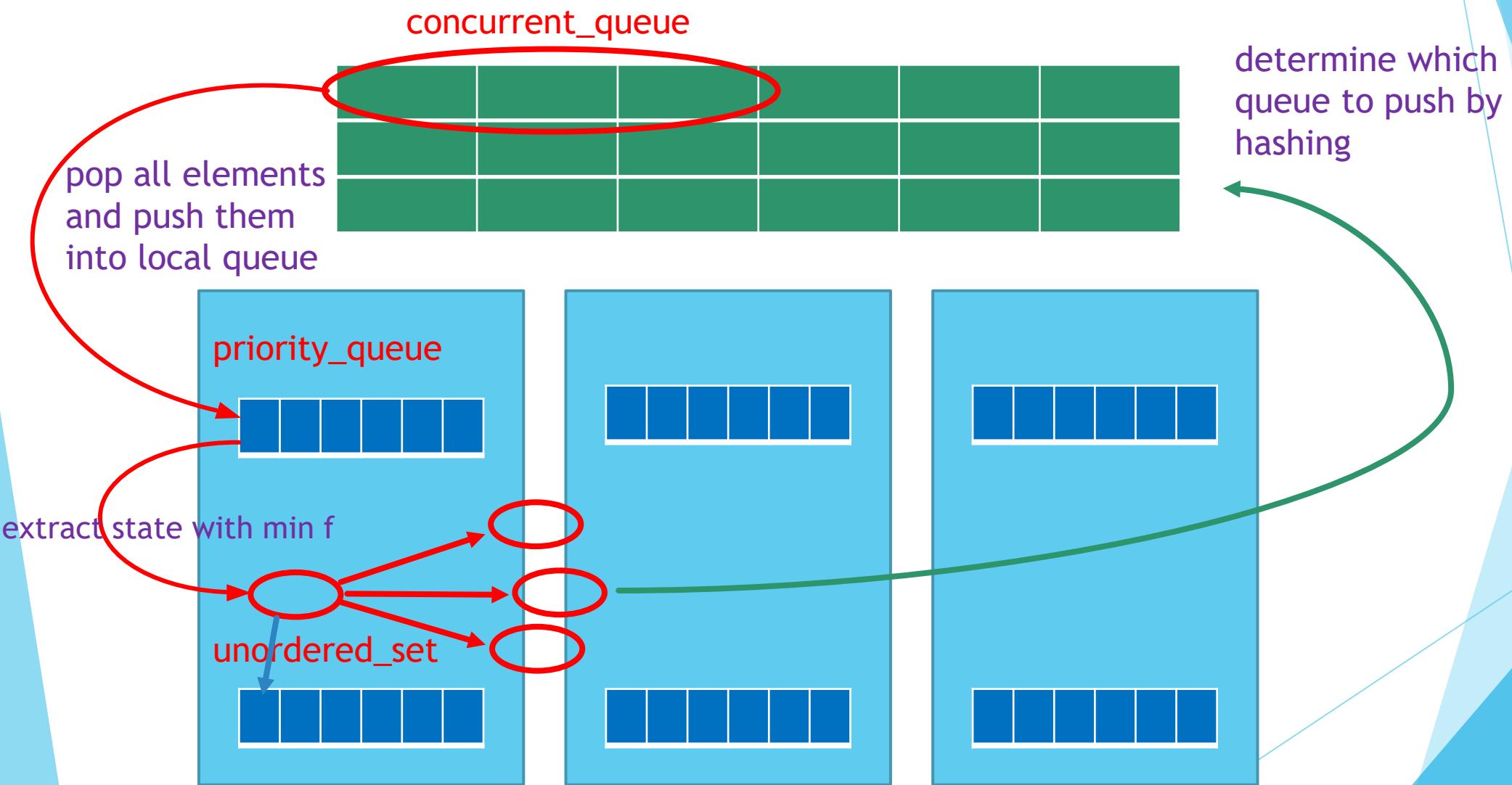
搜尋架構 - Parallel using OpenMP

- ▶ Hash Distributed A star (HDA*)
- ▶ Heuristic Calculation parallelized

A* State-Space Search



HDA*



Heuristic Calculation

```
for(int i=0; i<18; i++){  
    if(i % 3 == 0 || i % 3 == 2){  
        corners = corners + manhattan_distance(state, i, 0, true) + manhattan_distance(state, i, 2, true);  
        edges = edges + manhattan_distance(state, i, 1, false);  
    }else{  
        edges = edges + manhattan_distance(state, i, 0, false) + manhattan_distance(state, i, 2, false);  
    }  
}
```



```
#pragma omp parallel for reduction(+:corners, edges)  
for(int i=0; i<18; i+=3){  
    corners = corners + manhattan_distance(state, i, 0, true) + manhattan_distance(state, i, 2, true);  
    corners = corners + manhattan_distance(state, i+2, 0, true) + manhattan_distance(state, i+2, 2, true);  
    edges = edges + manhattan_distance(state, i, 1, false);  
    edges = edges + manhattan_distance(state, i+1, 0, false) + manhattan_distance(state, i+1, 2, false);  
    edges = edges + manhattan_distance(state, i+2, 1, false);  
}
```

Environment

產品集合

第 8 代 Intel® Core™ i7 處理器

代號

產品原名 Coffee Lake

垂直區段

Desktop

處理器編號

i7-8700

CPU 規格

核心數量

6

執行緒數量

12

最大超頻

4.60 GHz

Intel® 涡輪加速技術 2.0 頻率[†]

4.60 GHz

處理器基礎頻率

3.20 GHz

快取記憶體

12 MB Intel® Smart Cache

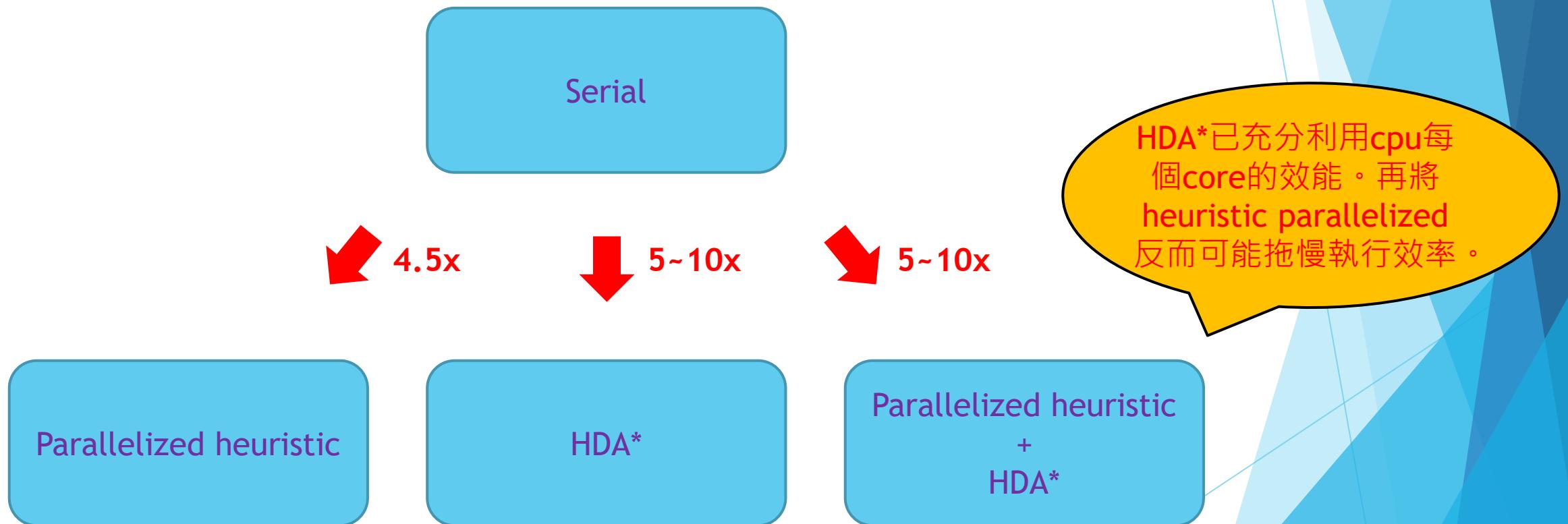
匯流排速度

8 GT/s

TDP

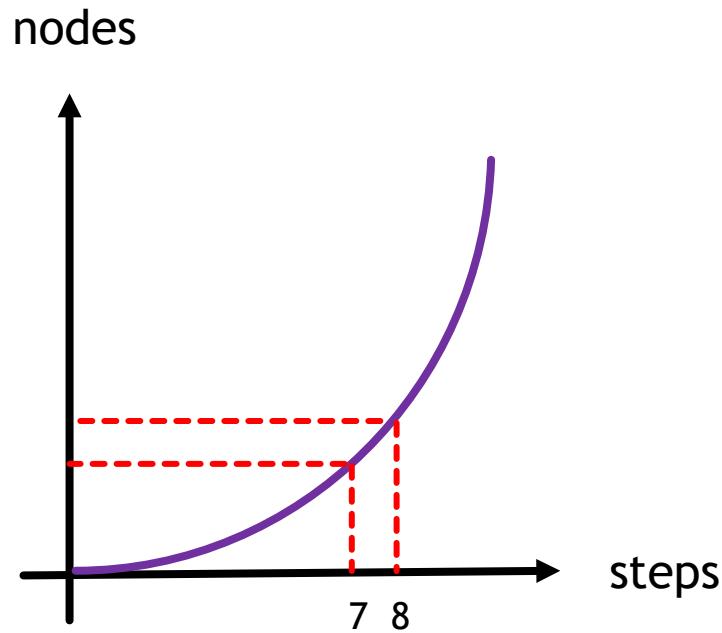
65 W

Speed-up



搜尋複雜度

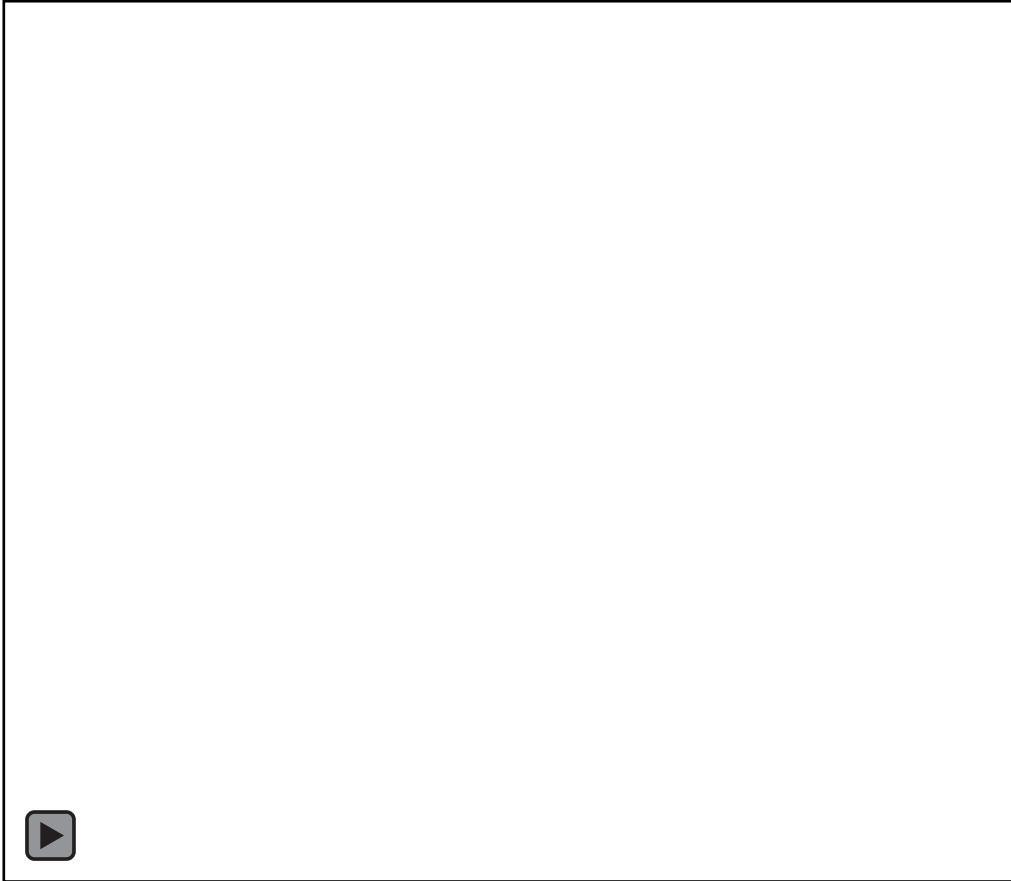
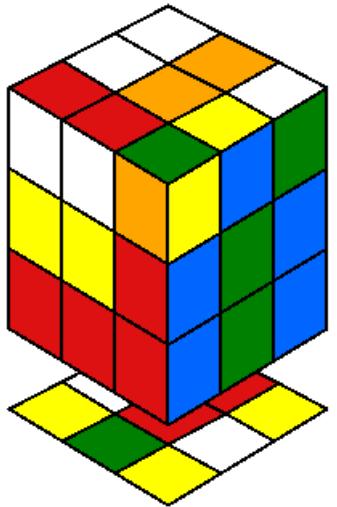
- ▶ 指數複雜度



Serial BFS search ,
Random generated 7-steps cases 10 times :

	min case	max case
num of node	1033390	7922840
execution time	13090.7 ms	86503.4 ms

測資 - 8步解

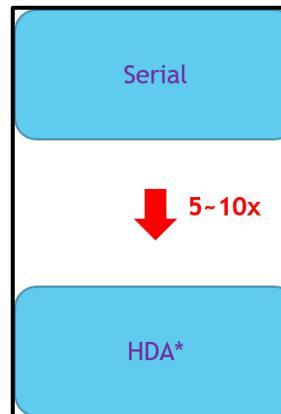


實驗結果

	BFS	A*	IDA*
serial	363247.479	40722.683	53366.825
parallel	36112.703	4203.660	7856.956

(ms)

↑
10x ↑
9.7x ↑
6.8x



平行化加速效果大致符合預期

分析

HDA* 優點

- ▶ 以global的concurrent queue作為每個thread的唯一溝通介面，存取效率高且維護成本低。
- ▶ 每個thread擁有自己local的open list和close list，運算過程獨立、不互相干擾、不需要同步機制。
- ▶ 以hash的方式決定thread產生出來的node要被push進哪個global queue，也就是決定將來這個node將被哪個thread執行。若hash得好可達成load balancing。

HDA* 缺點

- ▶ 每個thread擁有自己的close list且不互相溝通，可能導致同一個node被好幾個thread算到，增加許多不必要的運算，拖慢搜尋效率。
- ▶ 若hash得不好，可能導致某些global queue裡面有很多個node，而某些global queue為空。很多node的queue所對應到的thread loading很重；空的queue所對應到的thread就只能busy waiting等到有其他thread將node push進來。

更多步數實驗數據 (at least 20 times random generated case avg)

5 steps			
	BFS	A*	IDA*
serial	498.581	532.645	1184.61
parallel	530.978	115.363	226.403

node數太少，
直接BFS搜尋反而較快
(不須維持priority queue)

(ms)

6 steps			
	BFS	A*	IDA*
serial	6360.88	2817.07	6081.63
parallel	4660.1	625.258	991.442

(ms)

7 steps			
	BFS	A*	IDA*
serial	58937.9	23598.5	44935.9
parallel	50233.6	2687.57	4138.73

(ms)

更多步數實驗數據 (at least 20 times random generated case avg)

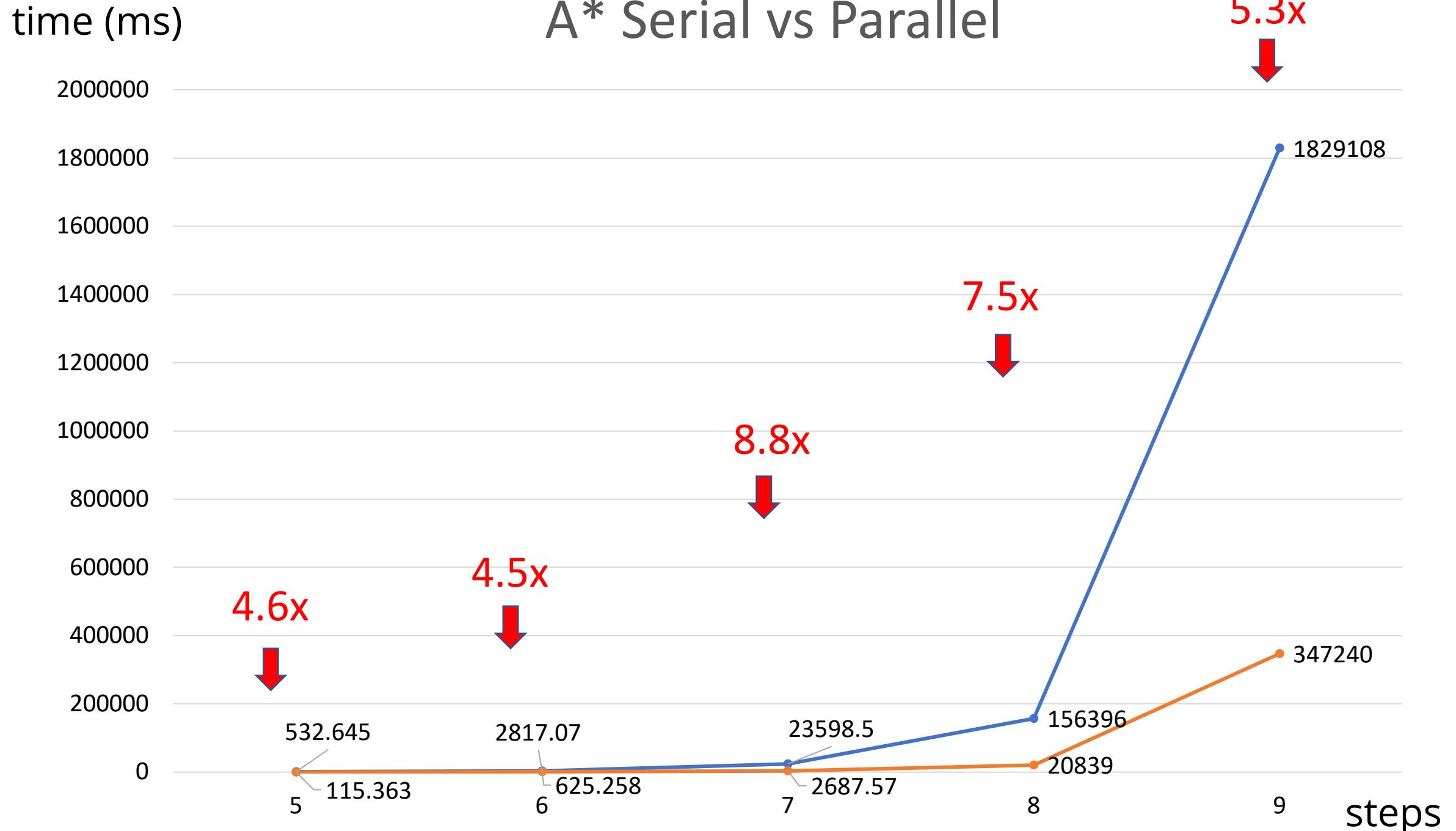
8 steps			
	BFS	A*	IDA*
serial	390757	156396	351065
parallel	195467	20839	54681

(ms)

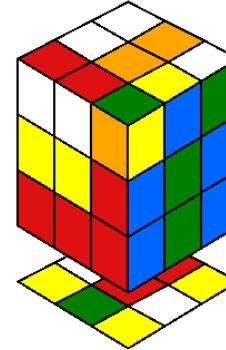
9 steps			
	BFS	A*	IDA*
serial	x	1829108	x
parallel	x	347240	792245

(ms)

A* Serial vs Parallel



HDA* using pthread vs OpenMP



(ms)

	BFS	A*	IDA*
pthread	32302.605	4309.505	7803.386
OpenMP	36112.703	4203.660	7856.956

Common Heuristic

- ▶ Manhattan distance
- ▶ Pattern database