Sparse Table

日月卦長

無修改區間操作

- 區間和
  用前綴和相減可以在 $O(1)$ 完成，預處理 $O(n)$

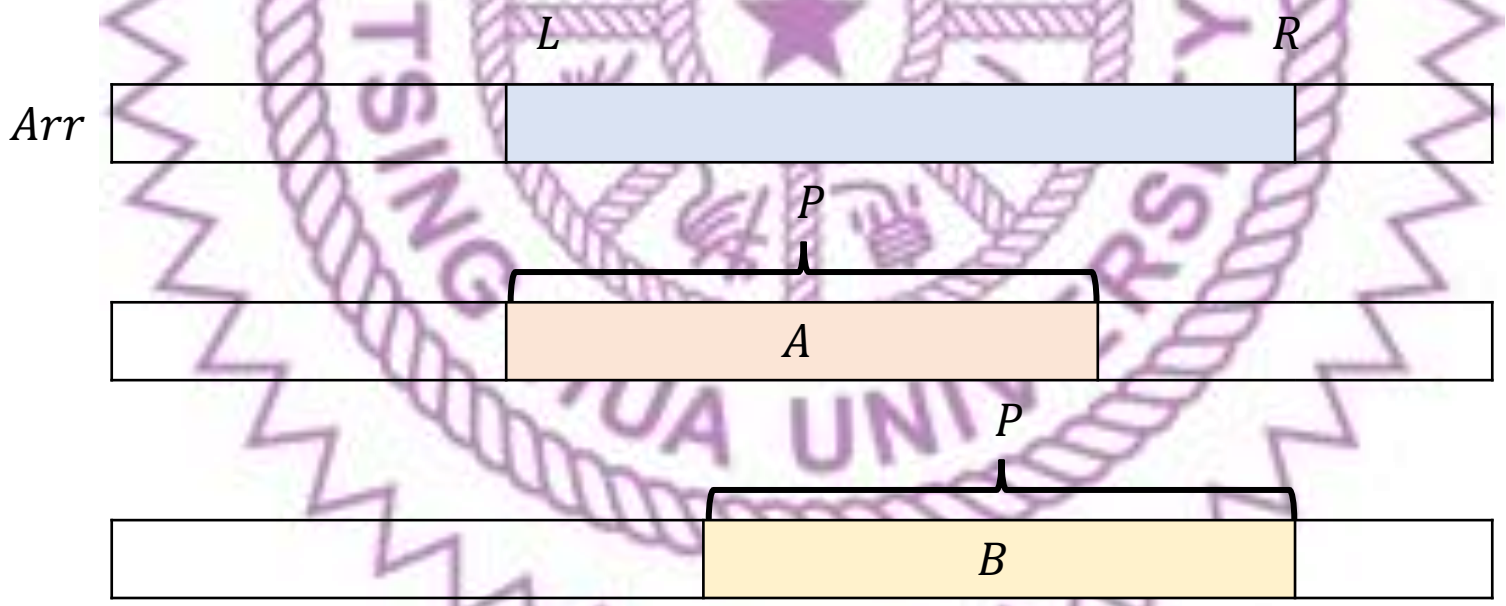
- 區間最大值、最小值想要 $O(1)$
  只能用 $O(n^2)$ 的時間和空間建表嗎？

區間最小值為例

- 設 $P \leq R - L + 1 \leq 2P$
- L 到 R 的最小值
  = min(A區域最小值, B 區域最小值)

只要 P 的所有可能
不要太多
就能建表 $O(1)$ 計算

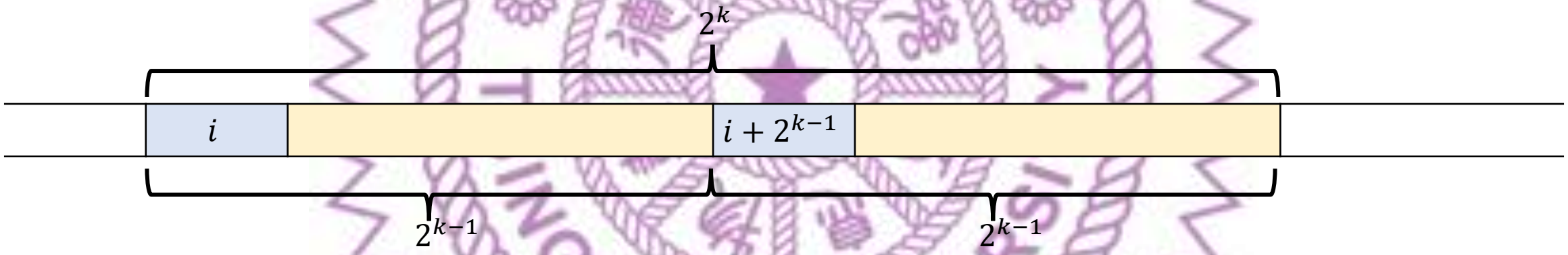
設 P 為
2^k



Sparse Table 倍增法建表

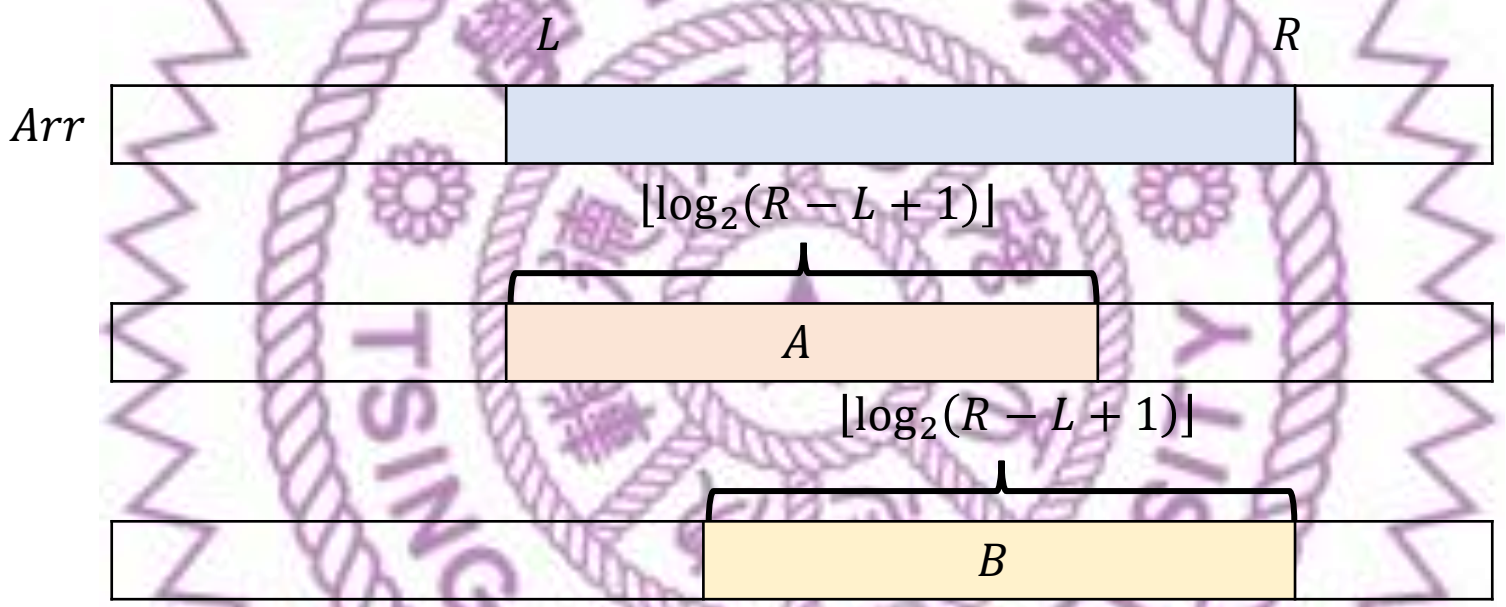
設 $Sparse_{k,i} = \min\{Arr_i, Arr_{i+1}, \ldots, Arr_{i+2^k}\}$

➡ $Sparse_{k,i} = \min(Sparse_{k-1,i}, Sparse_{k-1,i+2^{k-1}})$




```
void init() {
  for (int i = 0; i < n; ++i) Sparse[0][i] = Arr[i];
  for (int k = 1; (1 << k) <= n; ++k)
    for (int i = 0; i + (1 << k) <= n; ++i)
      Sparse[k][i] = min(Sparse[k - 1][i], Sparse[k - 1][i + (1 << (k - 1))]);
}
```

$O(1)$ 計算區間最小值




```
int query(int l, int r) {
  int k = std::__lg(r - l + 1); // O(1) 算 log 的黑魔法
  return min(Sparse[k][l], Sparse[k][r - (1 << k) + 1]);
}
```

完整程式碼

```
#define MAXN 1000000
#define MAXN_LOG 22
int n;
int Arr[MAXN];
int Sparse[MAXN_LOG][MAXN];
void init() {
  for (int i = 0; i < n; ++i) Sparse[0][i] = Arr[i];
  for (int k = 1; (1 << k) <= n; ++k)
    for (int i = 0; i + (1 << k) <= n; ++i)
      Sparse[k][i] = min(Sparse[k - 1][i], Sparse[k - 1][i + (1 << (k - 1))]);
}
int query(int l, int r) {
  int k = std::__lg(r - l + 1); // O(1) 算 log 的黑魔法
  return min(Sparse[k][l], Sparse[k][r - (1 << k) + 1]);
}
```

不喜歡黑魔法可以手動建 log 表 $O(n)$

```
int LogTable[MAXN + 1];
void init_log(int n) {
  int cur_log = 0;
  for (int i = 1; i <= n; ++i) {
    if (i == (2 << cur_log))
      ++cur_log;
    LogTable[i] = cur_log;
  }
}

int query(int l, int r) {
  int k = LogTable[r - l + 1]; // O(1)
  return min(Sparse[k][l], Sparse[k][r - (1 << k) + 1]);
}
```