

流程:

1. 於執行程式時讀入參數 Q ，決定整個 system 的 $space=Q+S$ 。
2. 根據不同的參數反覆做迴圈，每個 server 數量會產生 24 個 BP 及 Erlang 值。
3. 每個時間間隔 δ 會 rand(骰骰子)，共執行 $total\ time/\delta$ 次。
4. 迴圈內容：
 - (1) 每次 rand 出來的 x 跟 y 會分別跟 arrival probability 和 depature probability 比較，若小於該值則 arrival/depature。
 - (2) 若 arrival，arrival number +1，此時判斷系統 $space(Q+S)$ 是否已滿，滿了的話則 block；若 depature，server 正在服務的 number -1，其中 server 已滿的 probability 與 server 未滿的 probability 根據 server 服務的數量而有所不同。
 - (3) Arrival 且未 block、arrival 且 block、depature 等 3 種情況模擬完 $total\ time/\delta$ 次之後，將 arrival num/block num 則為其中 1 次 BP，重複 10 後得到平均，為一個 Blocking Probability。以不同 arrival、depature rate 分別算出 24 個平均 BP 與 Erlang 值(arrival rate/depature rate)。
 - (4) 根據不同 server 數量(1、5、10)跑出 table。
 - (5) 根據 $Q=0$ 、 $Q=S$ 跑出 2 table。
5. 由 table 觀察及直觀思考不難發現：server 越多(depature 機率更高)、Erlang 值越小(代表 depature 機率大於 arrival)，都容易使 blocking probability 更小。