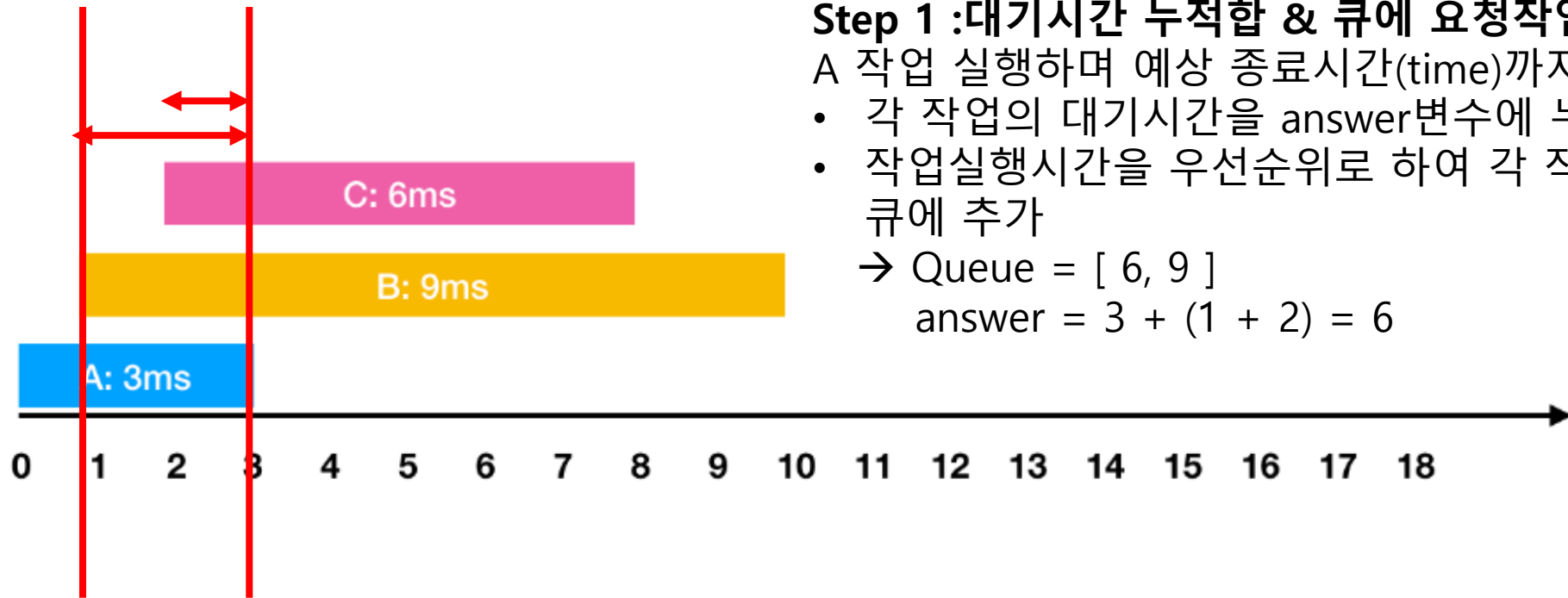


IDEA : 대기시간과 실행시간 누적합 후 평균



Step 1 : 대기시간 누적합 & 큐에 요청작업 추가

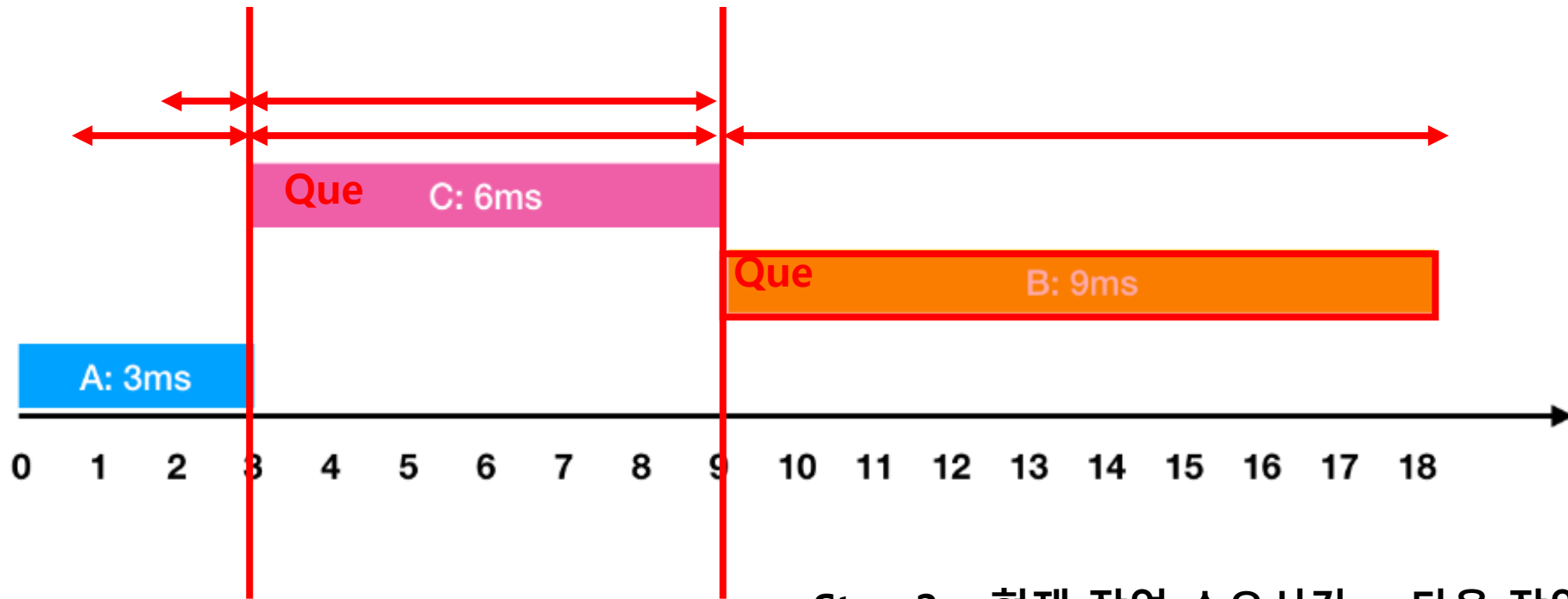
A 작업 실행하며 예상 종료시간(time)까지

- 각 작업의 대기시간을 answer 변수에 누적합
- 작업실행시간을 우선순위로 하여 각 작업 시간을 큐에 추가

→ Queue = [6, 9]

answer = 3 + (1 + 2) = 6

IDEA : 대기시간과 실행시간 누적합 후 평균

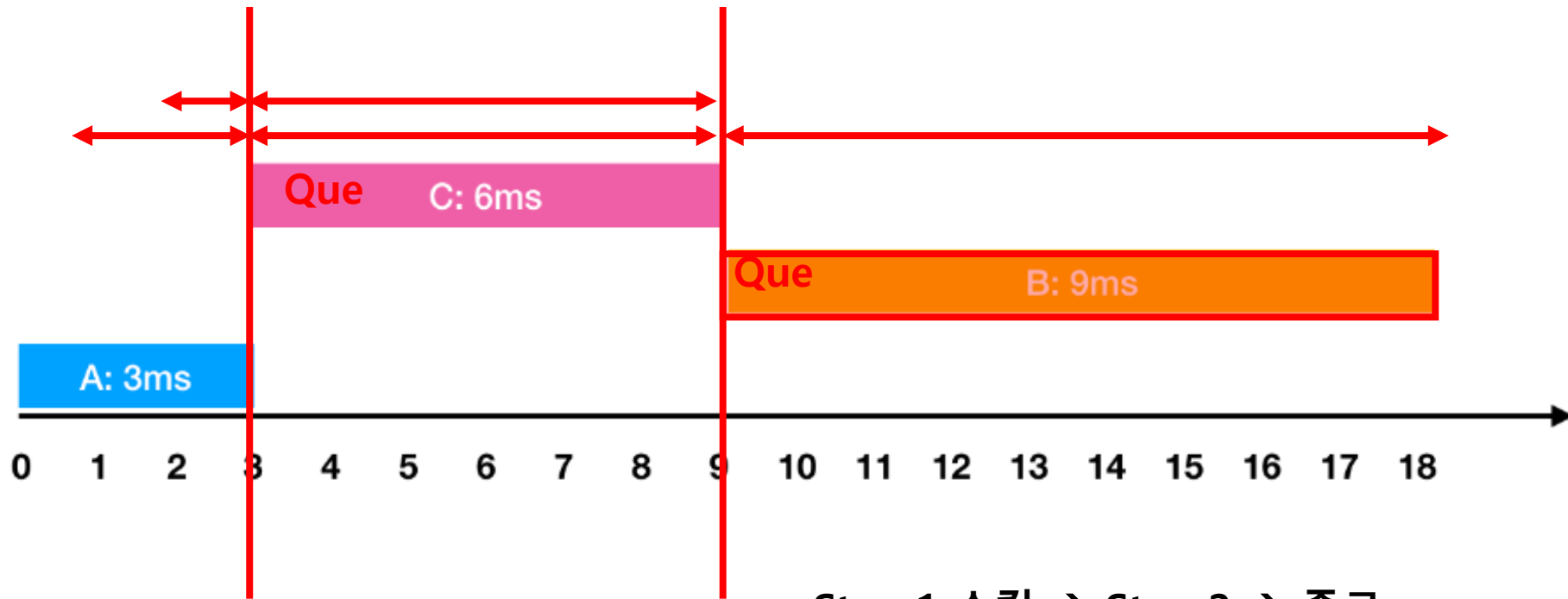


Step 2 :: 현재 작업 소요시간 + 다음 작업 소요시간

- 큐에 추가된 작업 중 가장 소요시간이 짧은 작업을 꺼내어 실행시간을 더함.
- 이 작업의 실행시간은 동시에 남은 작업의 대기시간이므로 (큐의 길이) * (짧은 작업의 대기시간)을 answer 변수에 누적합

→ Queue = [9]

IDEA : 대기시간과 실행시간 누적합 후 평균



Step 1 스킵 → Step 2 → 종료

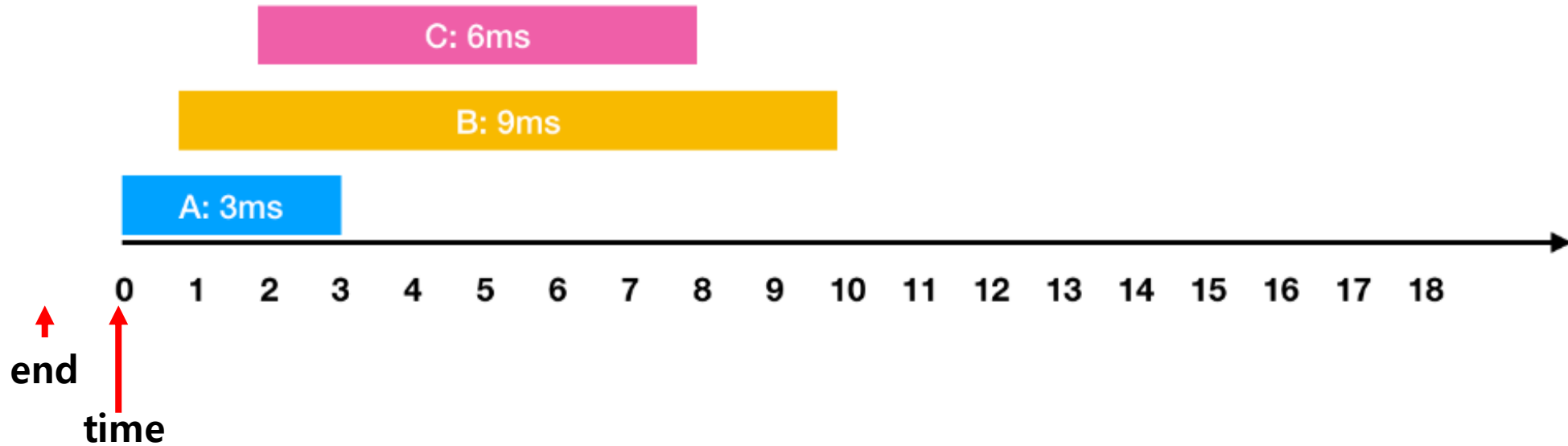
- 새로 추가할 작업이 없으므로 남은 작업 처리
- 누적합을 작업 개수로 나눠 요청부터 대기까지의 평균 시간을 구함

→ Queue = []

$$\text{answer} = (18 + 9) / 3 = 9$$

Step 1 :프로세스 대기시간 총합 / 우선순위 요청 작업 추가

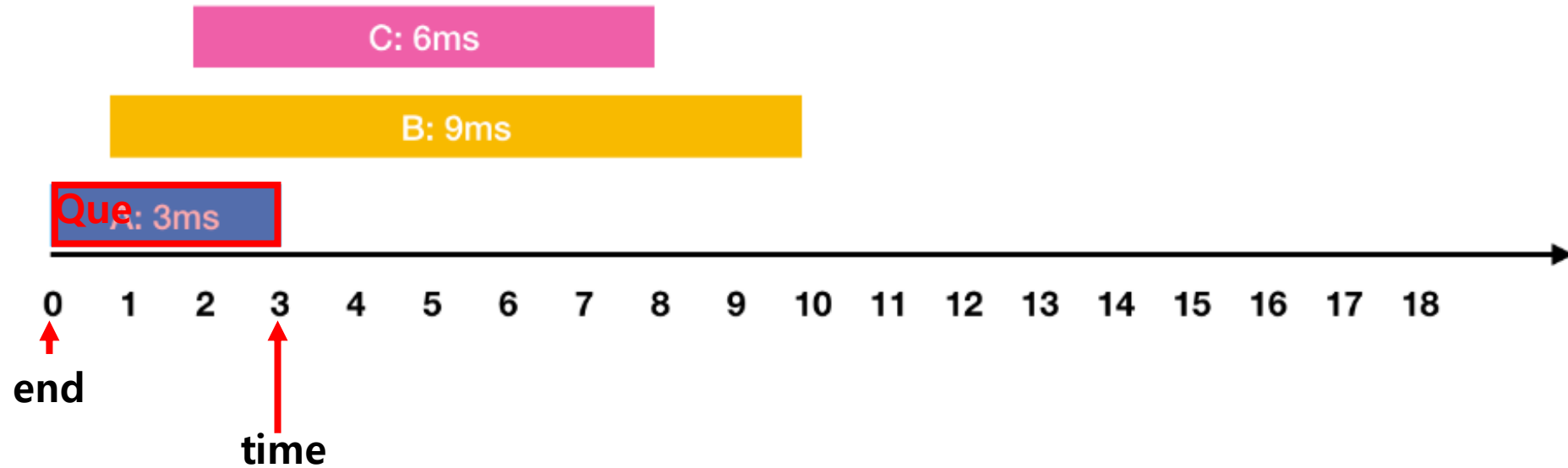
end < 요청시간 <= time 인 작업을 큐에 추가
최초 실행이므로 A가 선택되며
대기 시간 없음



count = 0
answer : 0
Queue : [3]

answer += (time - i[0])
→ 0 + (0-0) = 0

Step 2 : 최우선 작업 실행 시간 + 다음 작업 대기시간

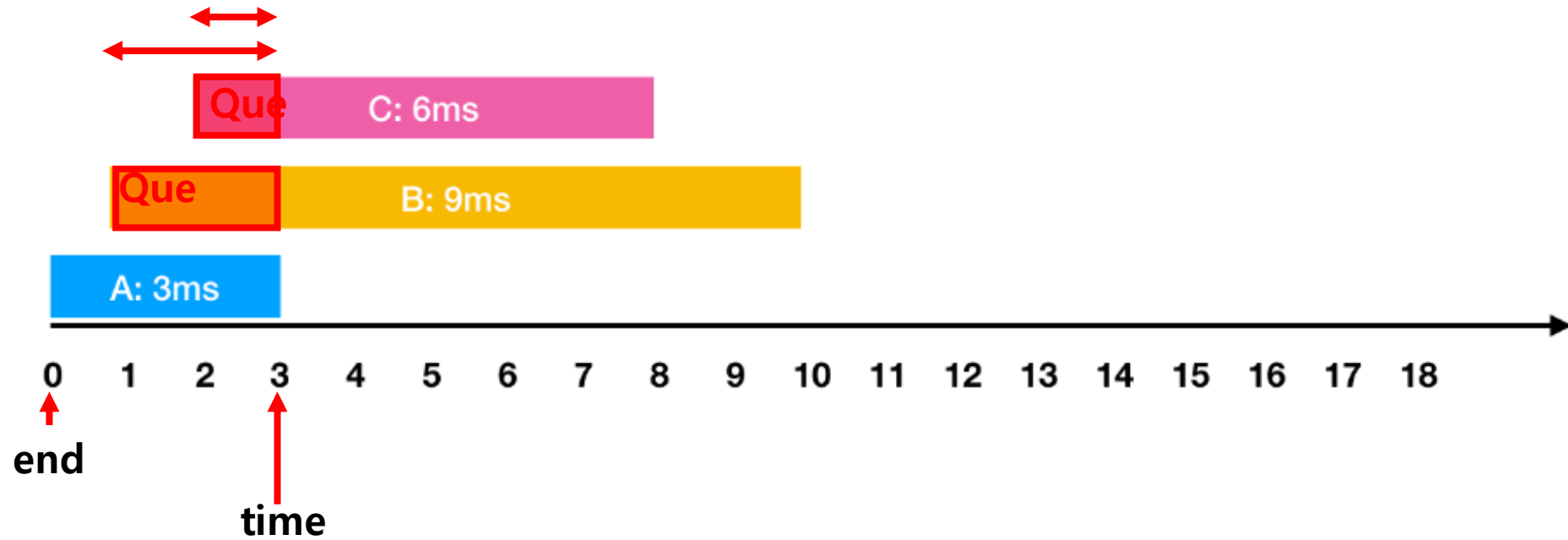


count = 1
answer : 3
Queue : []

answer += len(queue)*queue[0]
→ 0 + 1*3 = 3

Step 1

각 요청 작업의 대기시간을 모두 answer에 더함

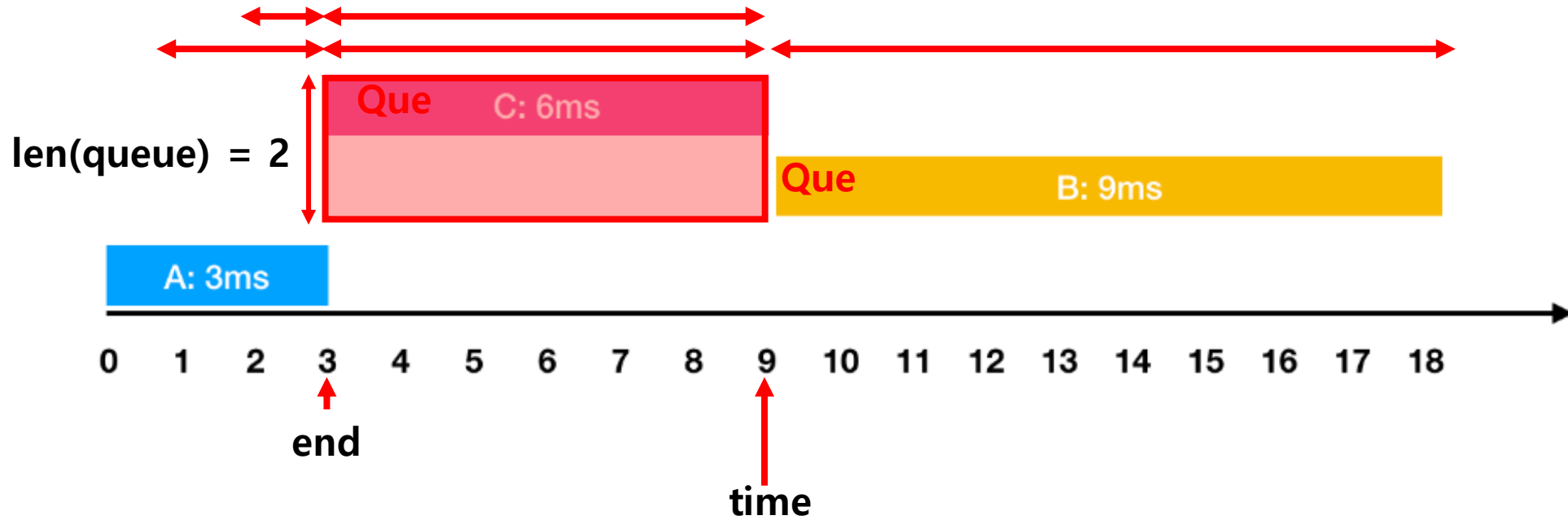


count = 1
answer : 6
Queue : [6, 9]

answer += (time - i[0])
→ 3 + (3-1) = 5
→ 5 + (3-2) = 6

Step 2

가장 빨리 끝나는 프로세스가 끝날 때까지
큐에 있는 프로세스 전부 대기시간 더함

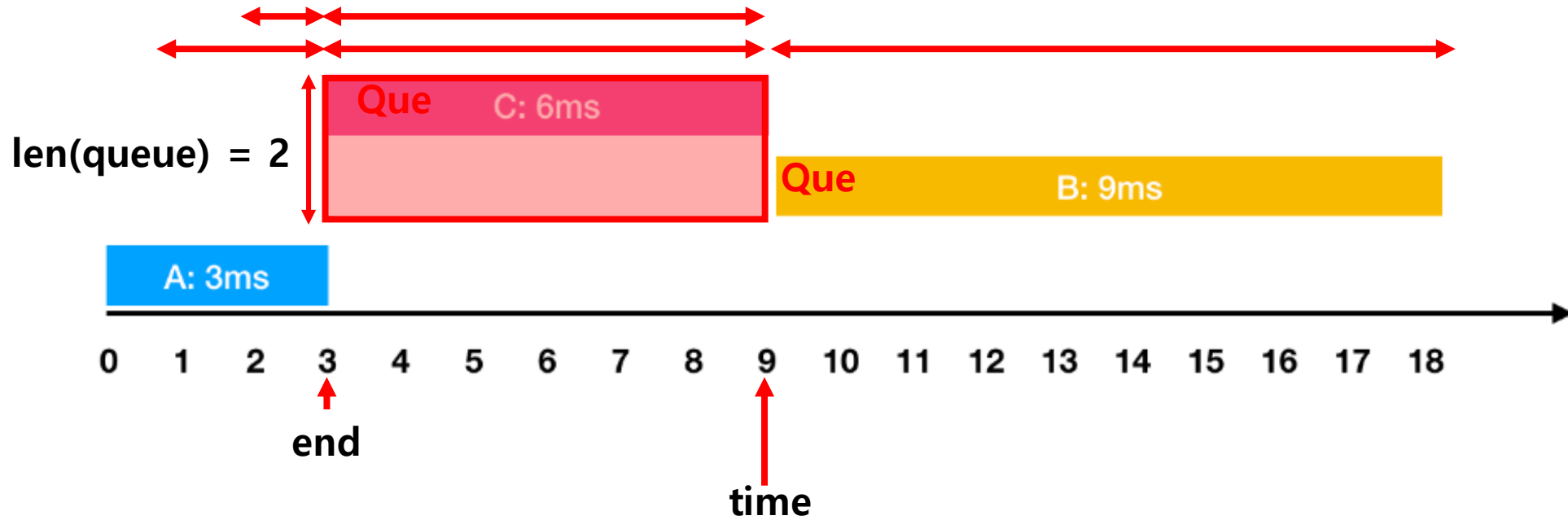


count = 2
answer : 18
Queue : [9]

answer += len(queue)*queue[0]
→ 6 + 2 * 6 = 18

Step 1

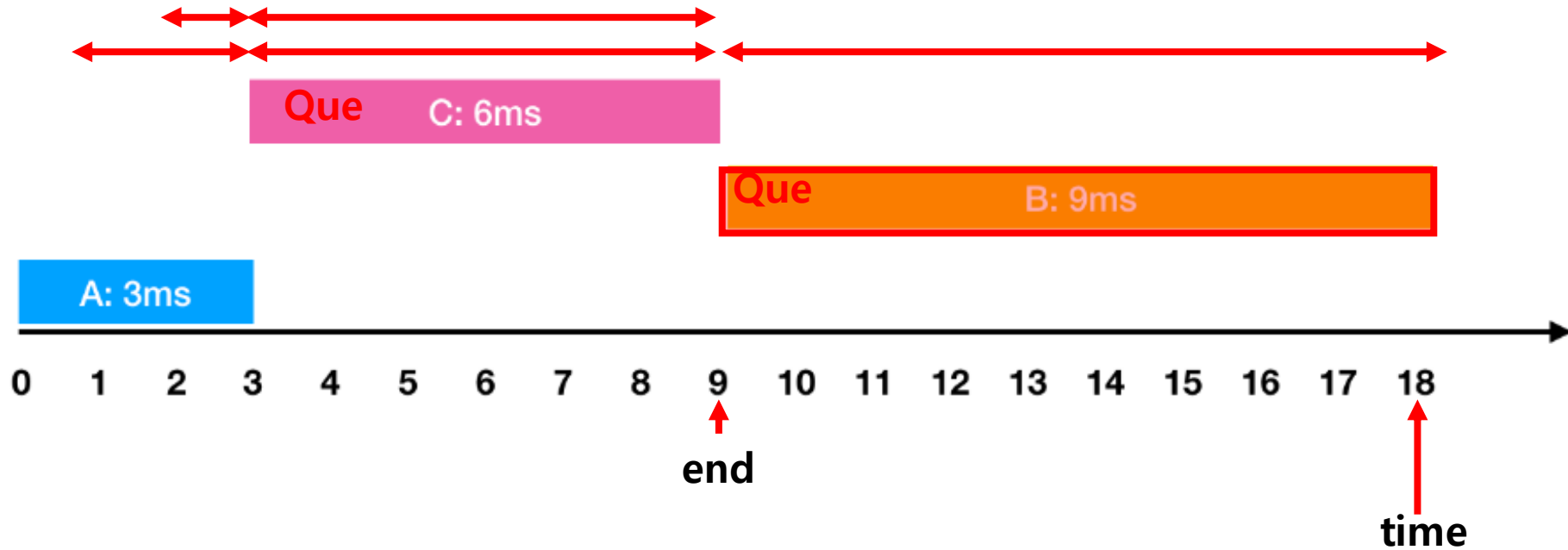
작업이 진행 중인 기간 동안 추가 요청 작업 없음
새로 들어올 경우 기존 큐에 있던 [2,9]와
작업소요시간 기준으로 비교해 정렬됨



count = 2
answer : 18
Queue : [9]

Step 2

작업이 진행 중인 기간 동안 추가 요청 작업 없음
새로 들어올 경우 기존 큐에 있던 [2,9]와
작업소요시간 기준으로 비교해 정렬됨

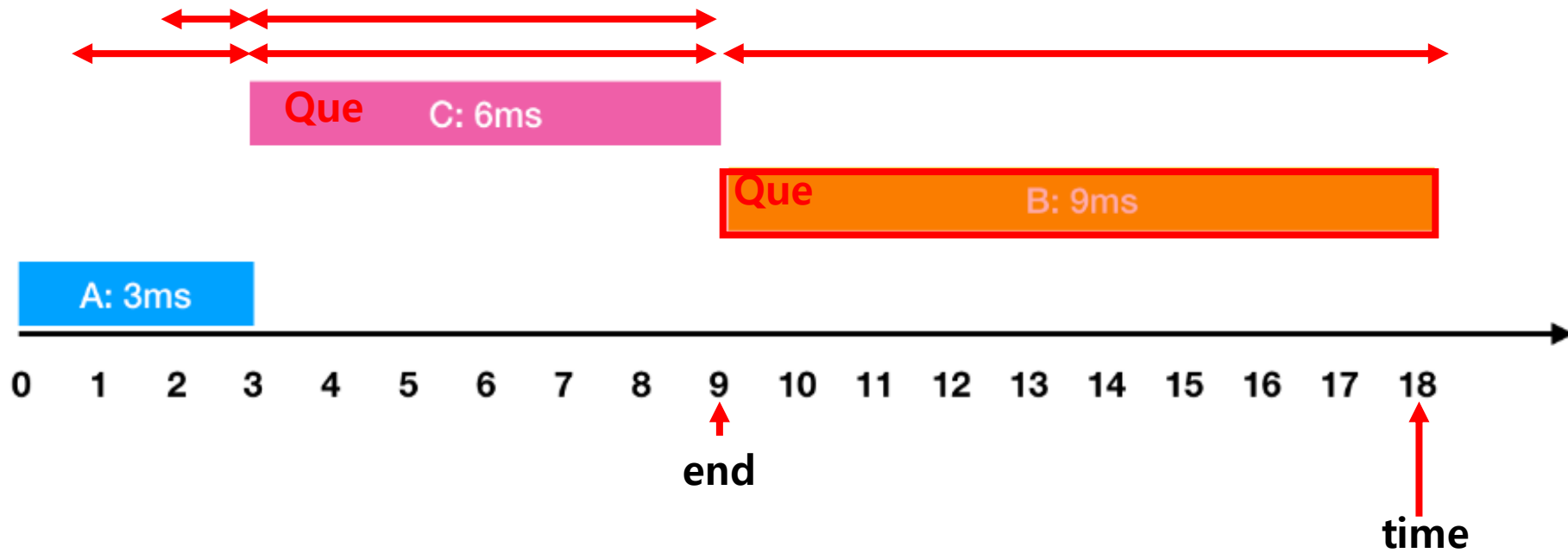


count = 3
answer : 27
Queue : []

answer += len(queue)*queue[0]
→ 18 + 1 * 9 = 27

Return answer

작업 요청부터 종료까지 소요 시간의 평균 구함



answer : $27 / 3 = 9$