**큐, 스택 정리**

10512김택서

만약 책을 내가 책을 본 순서대로 정렬한다 할때

제일 처음 본 책을 처음에,

제일 최근에 본 책을 마지막의 순서로 두어

오래된 책부터 다시 읽는 다면

이는 **큐**와 비슷한 자료구조라고 볼수 있다

아니면 반대로

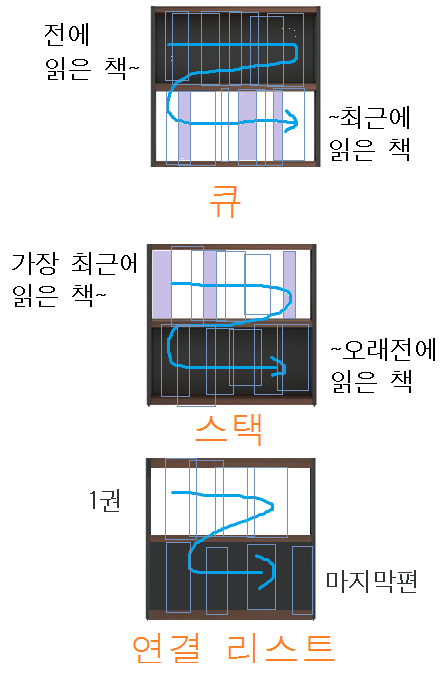
제일 처음 본 책을 마지막에,

 제일 최근에 본 책을

처음의 순서로 두어

최근 본책부터 다시 읽는 다면

이는 **스택**과 비슷한 자료구조라고 볼수 있다



다음부터는 이제 큐, 스택의 구조에 대해서 살펴본다

**----1.큐----**

**1-1.큐?**

queue. 뜻 **줄을 서서 기다리다**  
큐: 네트워크 운영체제 시간에 스케줄링기법에 대하여 배우면서

잠깐 봤었던 준비큐와 비슷하다

이제는 책보다는 영화를 예매하는 줄로 예를 들겠다

우리가 줄을 설때는 보통 선착순으로 선다

먼저 오면 앞에 줄을서고 그다음은 두에 그다음은 뒤에..

반대로 예매를 하고 난뒤 줄을 나갈때도 같다

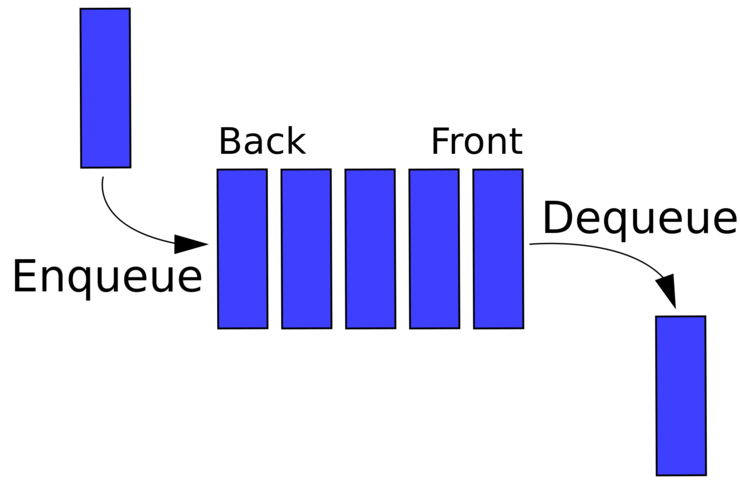
먼저 온 순서대로 나가고 뒤에있던 사람들은

앞으로 한칸씩 온다

이처럼 줄(메모리공간)을설때, 먼저 온 순서(입력되는 순서)대로

예매하려는 사람(입력되는 값)을 받는 방식을 큐 라고 한다

스케줄링 방법으로치면 FCFS(FIFO) 와 비슷하다



**1-2.큐의 친구들(?)**

큐를 사용할때는 몇가지의 기능들을 필요로한다

이는 큐가 하나의 함수가 아니라

프로그래머가 직접 구현하는 것이기 때문에

~~(여러 자료구조중에서 하나인건데 당연하지..)~~

큐는 하나의 개념일 뿐 그 개념을 구현하기 위해서는

필요한 기능들이 있다는것이다

큐는 문법이 아닌 문장이고

그 문장을 구성할 요소들이 필요한것과 같다

필요한 것은 크게 2가지로 나눌수 있는데

줄을 서게만드는 PUT (enqueue)의 기능

줄에서 빼주는 GET (dequeue)의 기능들이 있다

예제 소스

|  |  |  |
| --- | --- | --- |
| 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  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  64 | #include "studio.h"    int Q[5] = { 0, };  int i=0;    void enQ(int input)  {      if (i == 5)      {          printf("큐가 꽉 찼습니다");          return 0;      }      Q[i] = input;//입력된 값입력      i++;  }    void deQ()  {      int j;      Q[0] = 0;      for (j = 1; j < 5; j++)//앞으로 당김          Q[j - 1] = Q[j];      Q[4] = 0;      i--;  }    void show()  {      int j;      printf("큐의 저장상태\n\n");      for (j = 0; j < 5; j++)          printf("%2d", Q[j]);      printf("\n");      for (j = 0; j < 5; j++)          printf("%2d", j);      printf("\n");  }    int main(void)  {      int input, ennum;       while(1)      {          printf("1.큐에 값넣기\n");          printf("2.큐에 값빼기\n");          printf("3.큐상태보기\n");          printf("4.종료\n");          scanf("%d", &input);          system("cls");          if (input == 1)          {              printf("넣을 값 입력 : ");              scanf("%d", &ennum);              enQ(ennum);          }          else if (input == 2)              deQ();          else if (input == 3)              show();          else              return 0;      }    }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

이 예제는 큐에 값을 넣고 빼고 확인할 수 있게하는예제이다

Q라는 배열을 큐로 사용하고

함수 enQ는 수를 입력  
deQ는 첫번째를 0으로 만들고 하나씩 당김

show는 큐의 저장상태를 출력

/\*해서 하다\*/

큐는 주로 스케줄링이나 대기순서에 따라 처리하는 연산에서 쓰인다

**----2.스택----**

**2-1스택?**

stack 굴뚝

게임 stack 또는 스피드 스택을 생각하면 좀더 이해하기 쉽다

모두 '쌓는다' 는 개념에 공통적인 부분을 두고 있는데

자료구조의 스택도 비슷한 느낌이다

각각의 자료를 순서대로 쌓아 놓는다

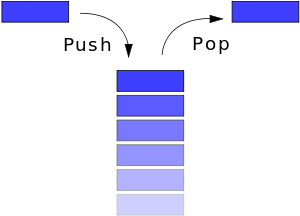
여기까지는 큐와 다른점이 없다

하지만 뜻은 굴뚝이란 뜻이있다

굴뚝에서는 연기가 하나의 방향으로만 나간다

스택의 형태도 비슷하다 자료가 들어온 구멍으로만 다시 나간다

마치 동전(자료)을 (메모리에)쌓은다음 다시 위에서부터 하나씩 빼는 것과 같다



**2-2 스택의 친구들**

스택은 큐와 자료구조가 다른만큼

큐와는 조금다른 친구들을 사귀고 있다

자료를 쌓는 push 자료를 뽑는 pop

이 두가지이다 push는 메모리에 차근차근 자료를 쌓는 역할을 하고

pop은 차근차근 쌓아 올린 자료들중에서 제일 상위에 있는 자료 하나를

슬쩍 빼 가져가는 역할이다

예제조스

|  |  |  |
| --- | --- | --- |
| 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  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 | #include "studio.h"    int stack[5] = { 0, };  int i=0;    int push(int input)  {      if (i >= 5)      {          printf("FULL!");          return 0;      }      else      {          stack[i] = input;          i++;      }  }    void pop()  {      if (i >= 0)      {          --i;          stack[i] = 0;      }  }    void show(void)  {      int j;      for (j = 4; j >= 0; j--)      {          printf("%d - %d \n", j + 1, stack[j]);      }  }    int main(void)  {      int input, stacknum;      for (;;)      {          printf("1. push\n");          printf("2. pop\n");          printf("3. 상태보기\n");          printf("4. exit\n");          scanf("%d", &input);          system("cls");          if (input == 1)          {              printf("값을 입력");              scanf("%d", &stacknum);              push(stacknum);          }          else if (input == 2)              pop();          else if (input == 3)              show();          else              return 0;      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

이 소스도 전의 큐와 같이 간단한 기능만 넣어놓았다

다른점이라면 자료의 구조를 큐에서 스택으로 바꾸었을 뿐이다

push에서는 입력받은 값을 쌓고

pop에서는 상위 자료만 없앤다

보기 이해하기 쉽도록 배열을 역순으로 출력하여

탑모양으로 출력되게 하였다

스택형식의 자료구조는 인터럽트를 처리할때 쓰인다