

templates
 splay trees
 exceptions

Q. Replace the guts of your priority queue, and replace them with a STL priority queue.

get
 insert
 isEmpty

```

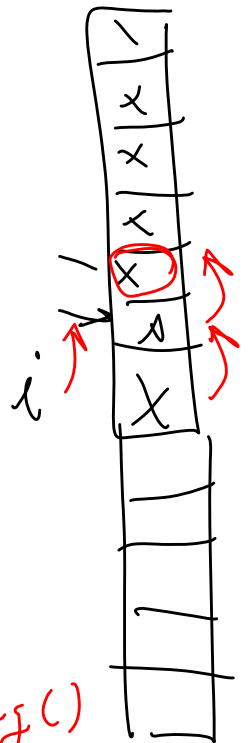
template < class typename T >
class PQ {
public:
  void insert ( T );
  T get ( );
private:
  T A[100];
  int i;
};
  
```

```

PQ::insert ( T x ) {
  A[i] = x;
  i++;
}
  
```

```

T PQ::get ( ) {
  i_min = 0;
  for j = 1 to i
    if ( A[j] < A[i_min] )
      i_min = j
  endfor
  return ( A[i_min] )
}
  
```



```
PQ <int> pq; //-- instance! yay!  
int x = 10;  
pq.insert(x);
```

```
PQ <Event*> queue;
```

class Foo

foo p, q;

```
bool operator < (Event* x, Event* y)
```

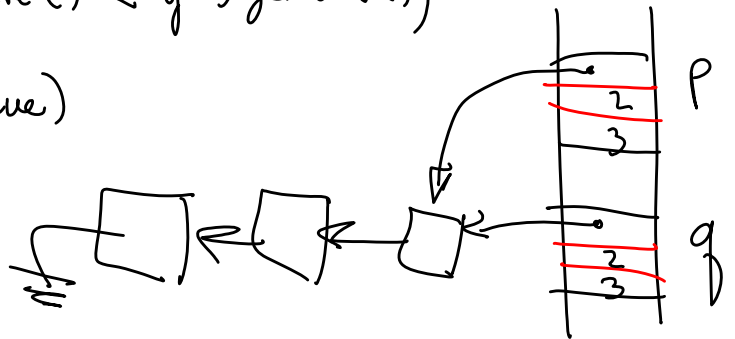
p = q;

```
if (x->getTime() < y->getTime())
```

```
return(true)
```

```
else
```

```
return(false)
```



```
class EventQ {
```

```
    Event * getEvent ( ) ;
```

```
    void insert ( Event * ) ;
```

```
    priority < Event * > queue ;
```

```
}
```

```
Event * EventQ :: getEvent ( ) {
```

```
    return ( queue . get ( ) ) ;
```

```
}
```