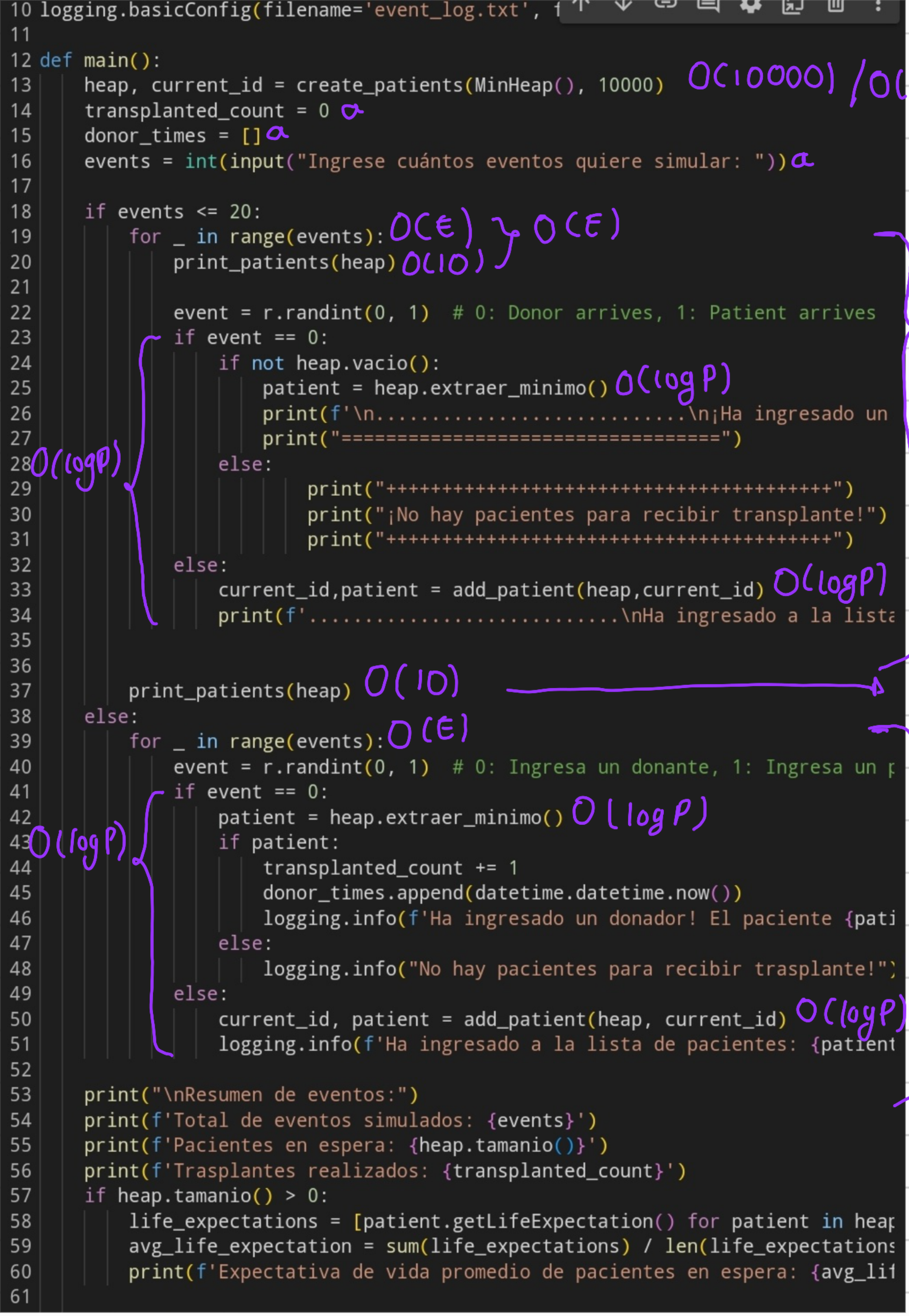

```
logging.basicConfig(filename='event_log.txt', f

def main():
    heap, current_id = create_patients(MinHeap(), 10000)
    transplanted_count = 0
    donor_times = []
    events = int(input("Ingrese cuántos eventos quiere simular: "))

    if events <= 20:
        for _ in range(events):
            print_patients(heap)

            event = r.randint(0, 1)  # 0: Donor arrives, 1: Patient arrives
            if event == 0:
                if not heap.vacio():
                    patient = heap.extraer_minimo()
                    print(f'\n...........................\n¡Ha ingresado un
                    print("=================================")
                else:
                        print("++++++++++++++++++++++++++++++++++++++++++")
                        print("¡No hay pacientes para recibir transplante!")
                        print("++++++++++++++++++++++++++++++++++++++++++")
            else:
                current_id,patient = add_patient(heap,current_id)
                print(f'...........................\nHa ingresado a la lista


        print_patients(heap)
    else:
        for _ in range(events):
            event = r.randint(0, 1)  # 0: Ingresa un donante, 1: Ingresa un p
            if event == 0:
                patient = heap.extraer_minimo()
                if patient:
                    transplanted_count += 1
                    donor_times.append(datetime.datetime.now())
                    logging.info(f'Ha ingresado un donador! El paciente {pati
                else:
                    logging.info("No hay pacientes para recibir trasplante!")
            else:
                current_id, patient = add_patient(heap, current_id)
                logging.info(f'Ha ingresado a la lista de pacientes: {patient

    print("\nResumen de eventos:")
    print(f'Total de eventos simulados: {events}')
    print(f'Pacientes en espera: {heap.tamanio()}')
    print(f'Trasplantes realizados: {transplanted_count}')
    if heap.tamanio() > 0:
        life_expectations = [patient.getLifeExpectation() for patient in heap
        avg_life_expectation = sum(life_expectations) / len(life_expectations
        print(f'Expectativa de vida promedio de pacientes en espera: {avg_lif

```

$O(10000)$ / $O(P)$

a

a

a

$O(E)$ $\}$ $O(E)$

$O(10)$

$O(\log P)$

$O(\log P)$

$O(\log P)$

Para el for:

$O(E + E\log P)$

$O(10)$ → $O(10)$

$O(E)$

$O(\log P)$

$O(\log P)$

$O(E\log P)$

$O(\log P)$

$O(P)$

Complejidad Temporal:

$$T_{(E,P)} = O(10000) + O(E + \log P) + O(P) + O(10)$$

$$T_{(E,P)} = O(10010 + E + \log P + P)$$