**Complejidad Computacional = N2 .** Debido a la anidación de 2 ciclos while con misma variable (N \* N).

**Caso#00: Enunciado**

*Descripción:* Se busca revisar el correcto funcionamiento del programa con el caso obtenido como ejemplo del enunciado

Caso#00.in Caso#00.out

|  |  |
| --- | --- |
| 10 100  42 2 50 10 1 50 24 18 23 | 3  6 |

**Caso#01: Empate en cantidad de aplicaciones contiguas**

*Descripción:* Se busca verificar que el programa devuelva el subconjunto con mayor tamaño al encontrar empate en cantidad de aplicaciones.

Caso#01.in Caso#01.out

|  |  |
| --- | --- |
| 7 100  50 10 40 20 35 50 9 2 10 | 3  4 |

**Caso#02: Empate en cantidad y tamaño de aplicaciones contiguas**

*Descripción:* Se busca verificar que el programa devuelva el primer subconjunto al encontrar empate en cantidad de aplicaciones y memoria.

Caso#02.in Caso#02.out

|  |  |
| --- | --- |
| 6 40  2 10 30 5 20 20 | 2  2 |

**Caso#03: Empate en cantidad y tamaño, con un número en común.**

*Descripción:* Se busca verificar que el programa devuelva el primer subconjunto al encontrar empate en cantidad de aplicaciones y memoria, siendo que comparten un número en común.

Caso#03.in Caso#03.out

|  |  |
| --- | --- |
| 5 50  10 30 20 30 5 | 2  2 |

**Caso#04: Cantidad de memoria insuficiente**

*Descripción:* Se busca verificar que el programa evalúe correctamente un caso en el cual la memoria no alcanza para instalar la aplicación deseada.

Caso#04.in Caso#04.out

|  |  |
| --- | --- |
| 5 100  20 20 19 20 20 | MEMORIA INSUFICIENTE |

**Caso#05: Elimina todas**

*Descripción:* Se busca verificar que el programa corrobore correctamente un caso en el cual se debe eliminar todas las aplicaciones existentes.

Caso#05.in Caso#05.out

|  |  |
| --- | --- |
| 6 710  200 150 100 50 200 10 | 6  1 |

**Caso#06: Respuesta óptima al final**

*Descripción:* Se busca verificar que el programa recorra toda la secuencia de aplicaciones instaladas antes de devolver el resultado.

Caso#06.in Caso#06.out

|  |  |
| --- | --- |
| 10 120  10 20 30 40 50 60 70 80 90 100 | 2  9 |

**Caso#07: Elimina solo una aplicación**

*Descripción:* Se busca verificar que el programa evalúe correctamente el caso más óptimo al luego de haber encontrado uno válido.

Caso#07.in Caso#07.out

|  |  |
| --- | --- |
| 6 60  100 80 30 30 60 80 | 1  1 |

**Caso#08: Elimina solo una aplicación, teniendo 2 opciones**

*Descripción:* Se busca verificar que el programa devuelva el primer caso óptimo encontrado.

Caso#08.in Caso#08.out

|  |  |
| --- | --- |
| 6 60  100 60 30 30 60 80 | 1  1 |

**Caso#09: Única aplicación**

*Descripción:* Se busca verificar que el programa verifique correctamente que hay una única aplicación a eliminar y que la misma alcanza para instalar la nueva.

Caso#09.in Caso#09.out

|  |  |
| --- | --- |
| 1 100  100 | 1  1 |