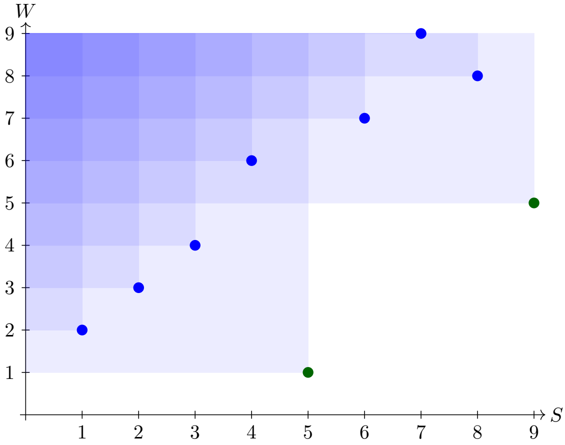
Solution for Aqueducts

Hugo Peyraud-Magnin

June 28, 2025

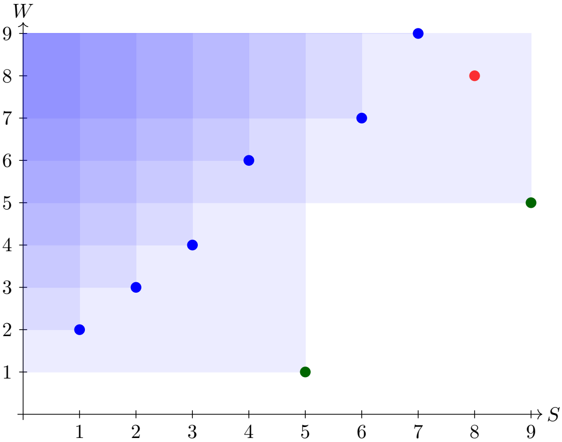
Statement

i dominates j iff $S_i > S_j$ and $W_i < W_j$.



Statement

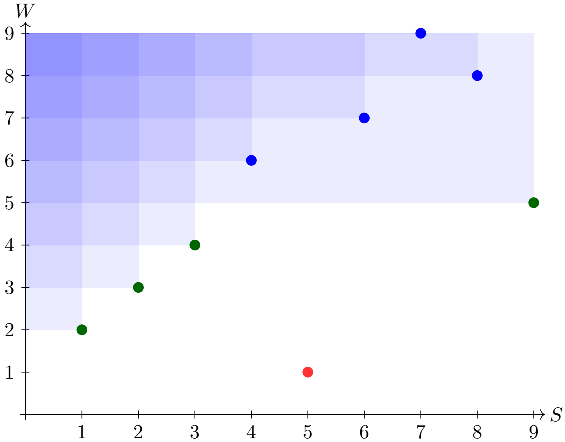
i dominates j iff $S_i > S_j$ and $W_i < W_j$.



For each material k, count the number of optimal (non-dominated) materials after removing k.

Statement

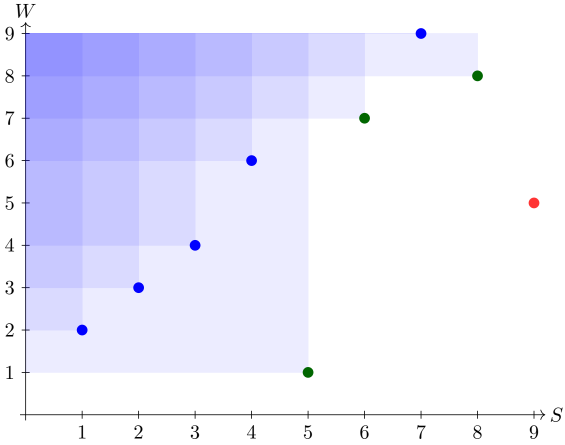
i dominates j iff $S_i > S_j$ and $W_i < W_j$.



For each material k, count the number of optimal (non-dominated) materials after removing k.

Statement

i dominates j iff $S_i > S_j$ and $W_i < W_j$.



For each material k, count the number of optimal (non-dominated) materials after removing k.

Subtask 1 ($N \leqslant 100$)

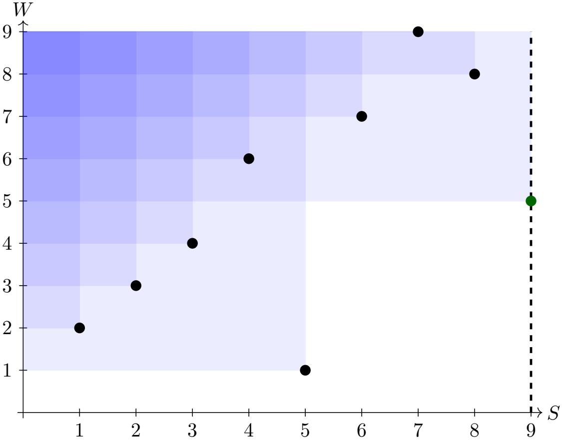
- For every k
  - For all $i, j \in \{0, 1, \ldots, N-1\} \setminus \{k\}$
    - If i dominates j, mark j as bad.
  - C_k is the number of unmarked materials.

For every k, computing C_k takes $\mathcal{O}(N^2)$. Overall time complexity: $\mathcal{O}(N^3)$.

Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.

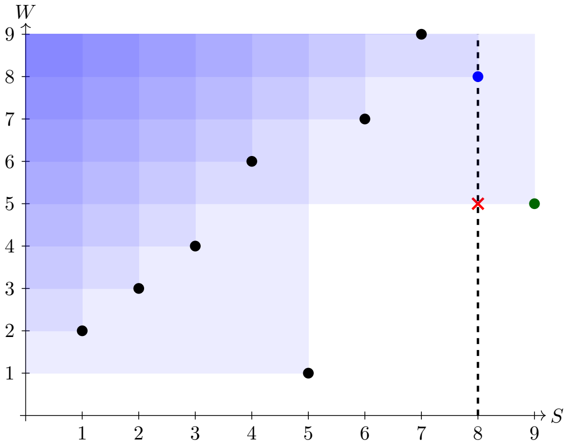
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

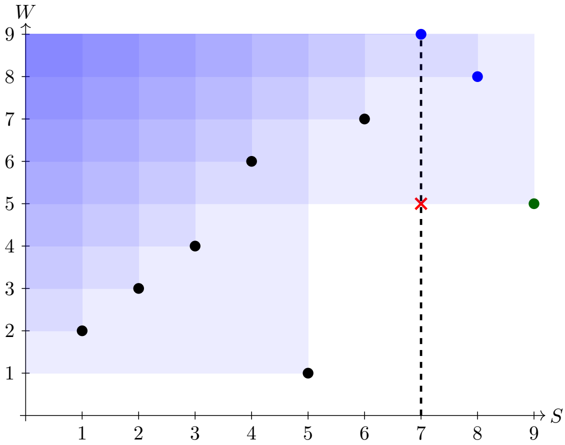
Observation: a material is optimal iff it is lower than every other material to its right.

Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



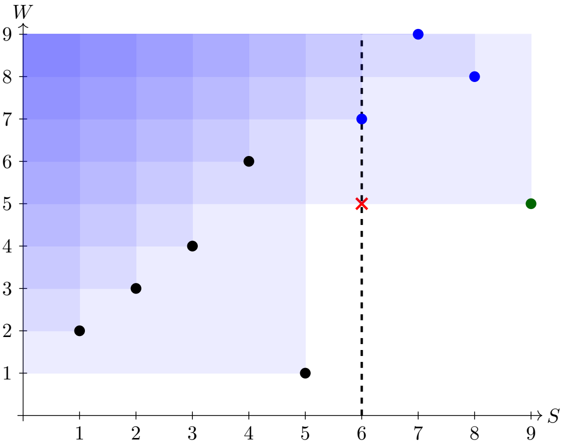
Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



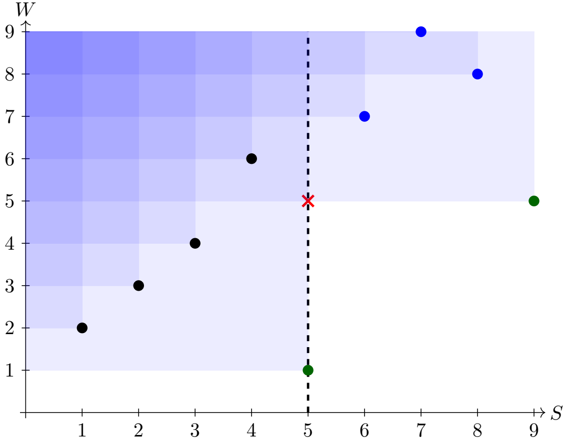
Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

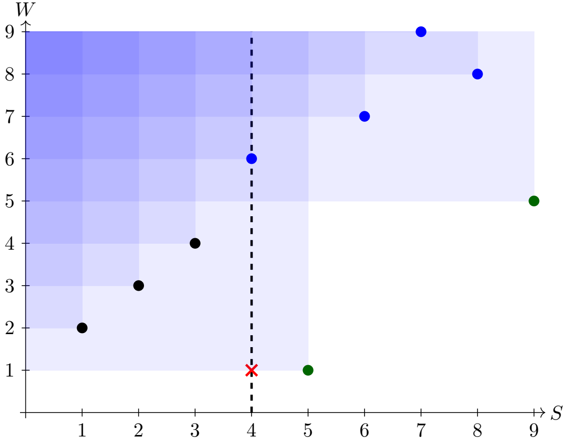
Observation: a material is optimal iff it is lower than every other material to its right.
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

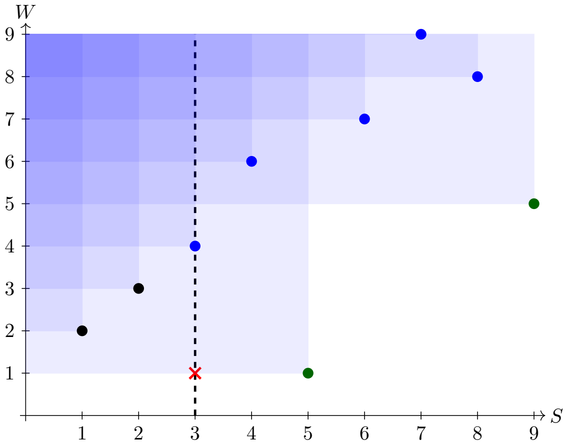
Observation: a material is optimal iff it is lower than every other material to its right.

Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

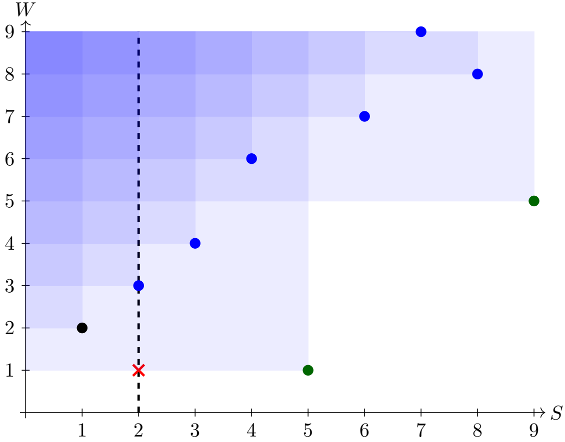
Observation: a material is optimal iff it is lower than every other material to its right.
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.

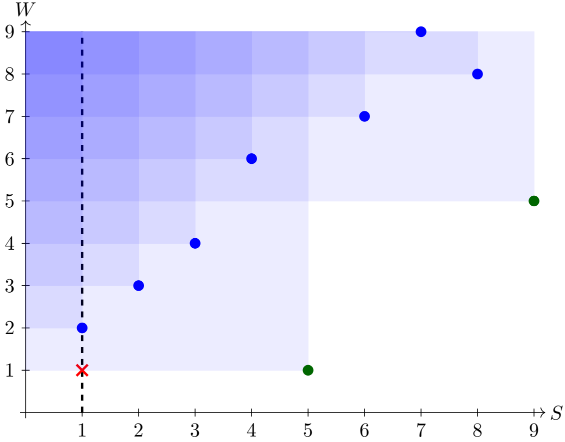
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.

Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.



Subtask 2 ($N \leqslant 5000$)

Observation: a material is optimal iff it is lower than every other material to its right.
Algorithm: iterate over materials sorted by decreasing strength, while maintaining the minimum weight.

First, sort the materials in $\mathcal{O}(N \log N)$. Then for every k, compute C_k in $\mathcal{O}(N)$. Total complexity: $\mathcal{O}(N^2)$.

Subtask 3 ($|S_i - W_i| \leqslant 5$)

Observation: i can only be dominated by materials j with $S_j \in \{S_i + 1, \ldots, S_i + 10\}$.

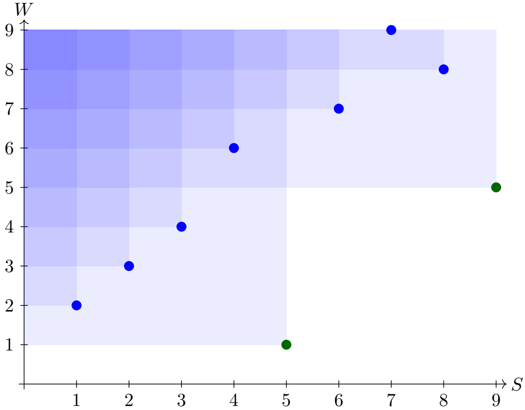
We can make an array or a map that associates each weight with the corresponding material (if it exists).

For every k, we can iterate over the (at most 10) materials dominated by k, and check for each of them if it's dominated by another material than k.

Subtask 4

Observation: Let T the number of optimal materials among the whole set of materials, and Δ_i the number of materials which are dominated only by i.

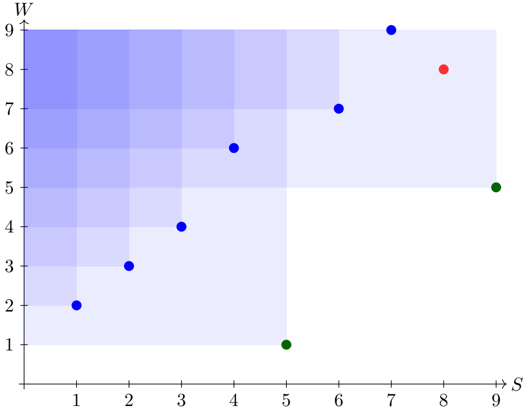
$$C_k = \begin{cases} T \text{ if } i \text{ was not optimal} \\ T - 1 + \Delta_k \text{ otherwise} \end{cases}$$



Subtask 4

Observation: Let T the number of optimal materials among the whole set of materials, and Δ_i the number of materials which are dominated only by i.

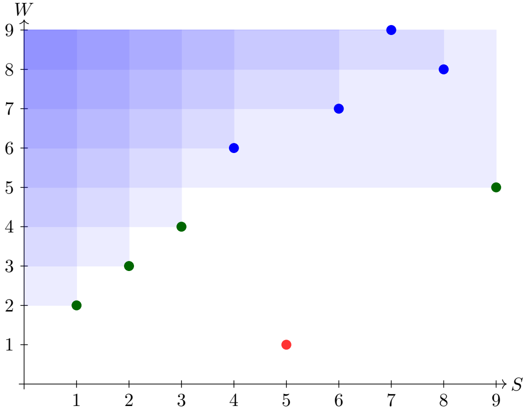
$$C_k = \begin{cases} T \text{ if } i \text{ was not optimal} \\ T - 1 + \Delta_k \text{ otherwise} \end{cases}$$



Subtask 4

Observation: Let T the number of optimal materials among the whole set of materials, and Δ_i the number of materials which are dominated only by i.

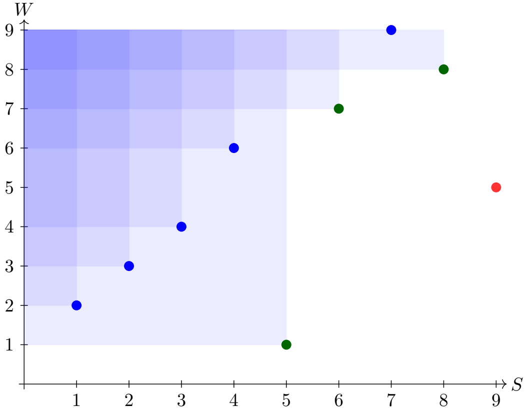
$$C_k = \begin{cases} T \text{ if } i \text{ was not optimal} \\ T - 1 + \Delta_k \text{ otherwise} \end{cases}$$



Subtask 4

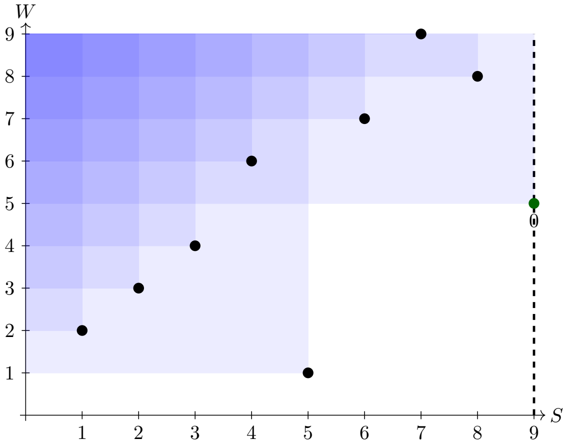
Observation: Let T the number of optimal materials among the whole set of materials, and Δ_i the number of materials which are dominated only by i.

$$C_k = \begin{cases} T \text{ if } i \text{ was not optimal} \\ T - 1 + \Delta_k \text{ otherwise} \end{cases}$$



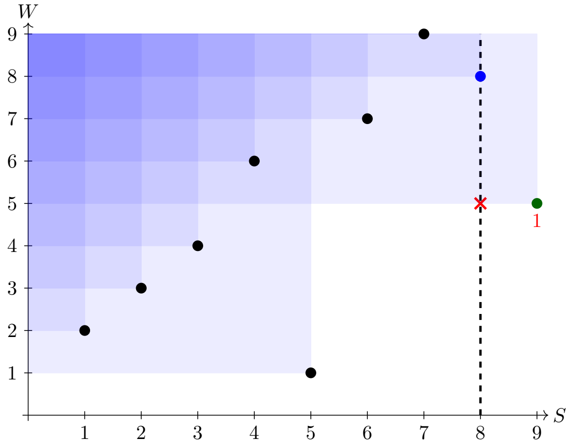
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and $i_{\min}$ the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\min}}$.



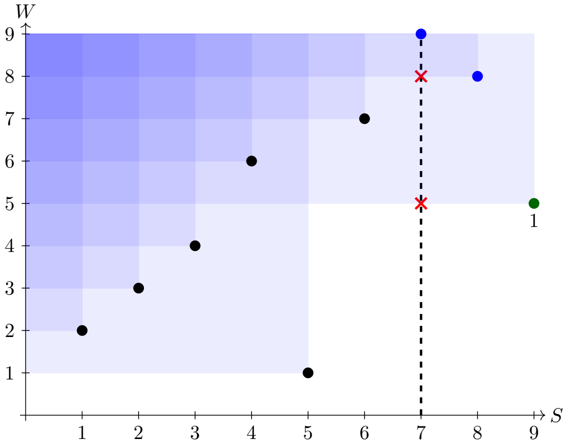
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and i_{min} the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\text{min}}}$.



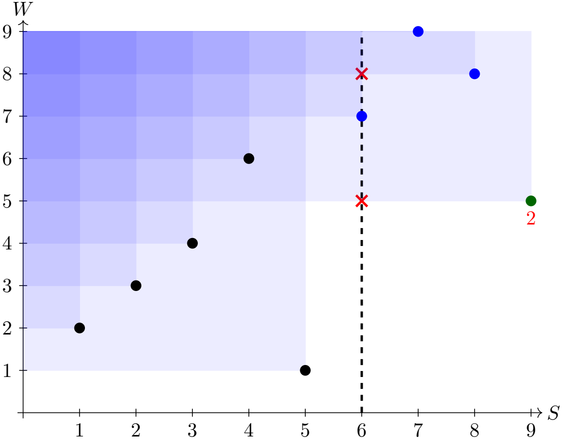
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and $i_{\min}$ the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\min}}$.



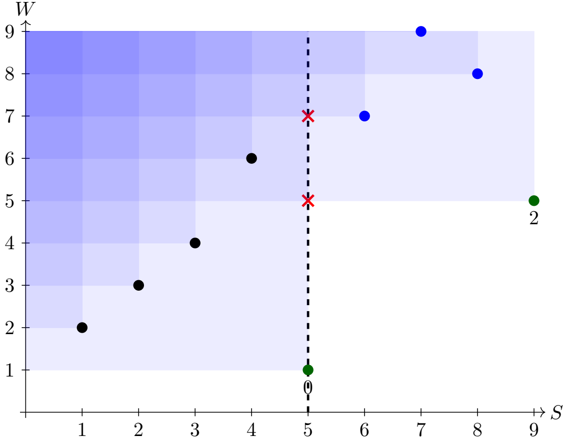
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and $i_{\min}$ the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\min}}$.



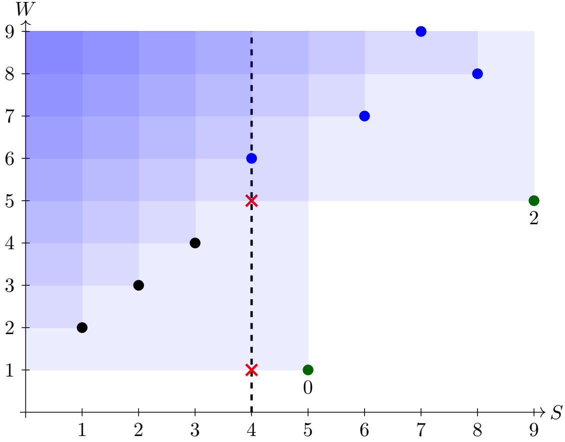
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and i_{min} the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\text{min}}}$.



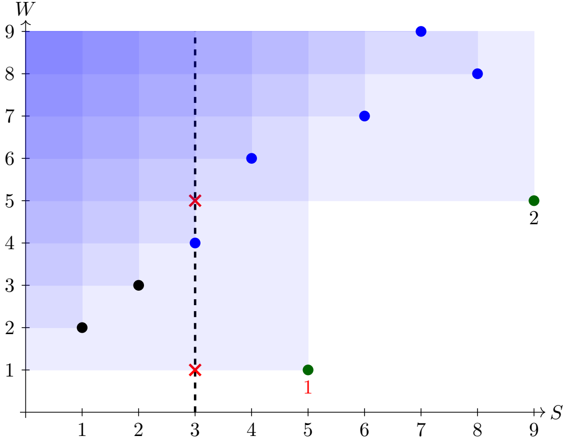
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and i_{min} the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\mathsf{min}}}$.



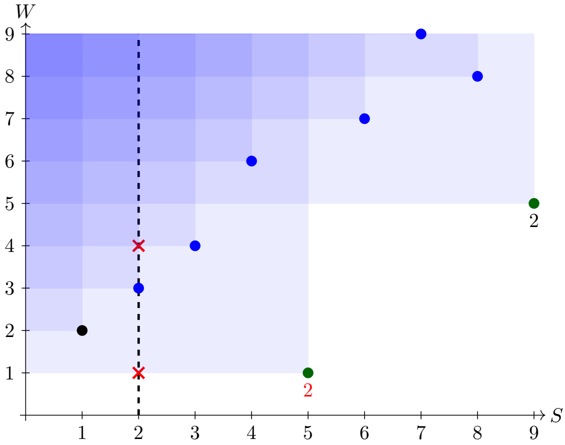
Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and $i_{\min}$ the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\min}}$.



Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and i_{min} the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\text{min}}}$.



Subtask 4

Algorithm: iterate over materials by decreasing weight, while maintaining the two minimum weights, and $i_{\min}$ the index of the material with minimum weight.
When the weight of the current material is between the two minimums, increment $\Delta_{i_{\min}}$. After sorting, it runs in $\mathcal{O}(N)$.