Zadanie 3.
Strukturą danych będzie wzbogacone drzewo AVL z gdzie klueczm jest liczba i.
Wzbogacenie drzewa polega na tym, że w każdym węźle trzymamy następujące dodatkowe parametry:

- wagę
- parę l, p(tak jak opisana w zadaniu)
- sumę wag elementów na przedziale od l do p
- krotkę knw w postaci (najwiekszy element w poddrzewie(nwe), najmniejsze takie l2 że to podrzewo zawiera wszystkie elemnty od l2 do nwe, suma wag od l2 do nwe(wnwe))
- krotkę knm w postaci (najmniejszy element w poddrzewie(nme), największe takie p2 że to podrzewo zawiera wszystkie elemnty od nme do p2, suma wag od nme do p2(wnme))

Przy takiej strukturze operacje Interval i Weight sprowadzają się do znalezienie elementu i w drzewie, jako że jest to drzewo AVL ich złożonośc to O(log|S|) czasowa i O(1) pamięciowa.
Operacja Ini to inicjacja pustego drzewa AVL, co ma złożoność O(1) czasowa i pamięciową.
Operacja Insert jeżeli i nie ma w zbiorze, wstawia i do zbioru a następnie liczy ponownie parametry węzłów, które wykonały rotacje; swoich prodków; i ustawia swoja w następujący sposób:

- ustawia l i p na siebie samego, wagę na swoją wagę i sumę wag na swoją wagę
- jeżeli nme prawego dziecka jest równe i + 1, ustawia p na p2 prawego dziecka oraz do sumy wag dodaje wnme, analogicznie jeżeli nwe prawego dziecki jest równe i-1, ustawia l na l2 lewego dziecka oraz do sumy wag dodaje wnme.
- Jeżeli jest liściem ustawia obie pozostałe krotki na (i, i , w), wpp:
  - kmw ustawia na kmw swojego prawego dziecka rozszerzoną o ile l2 prawego dziecka = i+1 o siebie i jeżeli nwe lewego dziecka to i-1 to o knw lewego dziecka
  - kme analogicznie

Jeżeli i było już w zbiorze znajdujemy przypisany mu wezeł i aktualizujemy jego parametry dotyczące wag w opisany wyżej sposób oraz robimy to samo z jego przzodkami.

Zauważmy, że aktualizacja parametrów w węźle wykonuję się w czasie stałym. A liczba „dotkniętych” wezłów jest taka jak przy zwykłym insercie w drzewie AVL więc koszt Inseret jest O(log|S|) czasowy i O(1) pamięciowy.