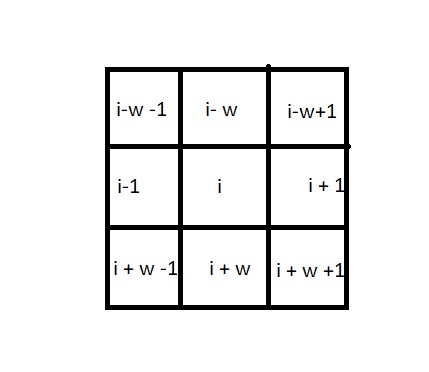
**Logik Minröjning**

* Varje Square-objekt sparas i en lista. Om spelplanen är 8x8 så är lista *squares* en lista med 64 objekt.
* Varje Square-objekt har ett intervall (en lista) av koordinater som den existerar på (i detta fallet är en ruta 25x25 pixlar). Om användaren trycker på en viss koordinat på spelplanen itereras listan *squares* tills den finner ett objekt som täcker muspositionens x- och y-intervall. Programmet vet då vilket objekt som det gäller.
* För att slumpa ut minorna så skapas en lista *squares\_i*, som är en lista med varje index i listan *squares*. *squares\_i*[0] *= squares*[0], *squares\_i*[1] *= squares*[1] osv.Programmet slumpar sedan ett tal *random\_int* mellan 0 och len(*squares\_i*). *squares\_i*[*random\_int*] flyttas från *squares\_i* till (raderas från listan och läggs till i) *squares\_w\_mines*. Alla objekt i *squares\_w\_mines* har alltså minor i sig. Samma process upprepas tills alla minor har blivit utplacerade.
* För att känna av om en ruta har några närliggande rutor som innehåller minor används följande illustration, där *i* är indexet i listan *squares,* och *w* är spelplanens bredd.

Specialfall görs för rutorna på kanterna och hörnen.

* När spelaren klickar på en ruta som innehåller 0 närliggande minor så öppnas också närliggande rutor med 0 närliggande minor med rekursion. Detta sker genom att närliggande rutor som inte redan är öppna till en kö, *queue*. Programmet öppnar sedan alla rutor i *queue.* Om rutan i *queue,* innehåller 0 närliggande minor så kallas funktionen igen, och den körs till att ingen ruta har en närliggande ruta som i sin tur har 0 närliggande minor. Till skillnad från punkten innan så kollar programmet bara i fyra riktingar (nord, syd, väst, öst) istället för åtta.