



4. Een muur neemt de plaats van 1 vakje in.
5. Vakjes liggen in dezelfde kamer indien je vanuit het ene vakje via horizontale of verticale verplaatsingen het ander vakje kan bereiken zonder een muur tegen te komen. Diagonale verplaatsingen zijn niet toegelaten.

Een voorbeeld van een willekeurige verdeling in kamers van een opgegeven gebouw is afgebeeld op de eerste pagina. Muren worden voorgesteld door een X.

## 2 Implementatie

1. Schrijf een verdeel-en-heers-algoritme voor het bovenstaand probleem. Het probleem kan initieel duidelijk opgedeeld worden in twee gelijkwaardige delen. Deel de grote ruimte eerst in twee willekeurige kamers in. Verdeel elk van deze kamers opnieuw in twee willekeurige kamers en ga zo recursief verder. Een kamer die bestaat uit minder dan  $d + 1$  vakjes wordt niet verder opgedeeld.
2. Implementeer je verdeel-en-heers-algoritme door de Java-code van het bestand *RoomBuilder.java* aan te vullen. Het is niet toegelaten de headers van de publieke methoden te wijzigen.
3. Het Java-bestand *Building.java* mag niet gewijzigd worden. Dit bestand zorgt voor een door ons vastgelegd uitvoerformaat.

## 3 Indienen

- Java-bestanden:

1. Het Java-bestand *RoomBuilder.java* moet via MINERVA onder de rubriek Studentenpublicaties geplaatst worden. Dit ten laatste op dinsdag 21 maart 2006 om 17 uur.
2. Vermeld in elk Java-bestand je naam in commentaar.

- Verslag:

1. Vermeld bovenaan je verslag je naam en familienaam !!!
2. Beschrijf in dit verslag de werking van je algoritme (pseudocode algoritme + uitleg bij pseudocode).
3. Bij voorkeur wordt het verslag opgesteld in L<sup>A</sup>T<sub>E</sub>X.
4. Je verslag geef je af in afgedrukte vorm tijdens het practicum Algoritmen en Datastructuren I op donderdag 23 maart 2006 om 17 uur.