

Tweede serie sommen.

1. Gegeven is een graaf met gewichten en twee knopen s en t in die graaf. Sommige kanten kunnen weggenomen worden zonder dat het kortste pad van s naar t verandert. Misschien is er een kant in de graaf zodat er na wegnemen van die kant *geen* pad meer is van s naar t . Beschrijf een efficiënte algoritme die de kant selecteert met de eigenschap dat na wegnemen van die kant de lengte van het kortste pad van s naar t *zoveel mogelijk* toeneemt. Wat is de complexiteit van uw algoritme?
2. Een skischool heeft n leerlingen en n paren skis. De regel is dat de lengte van de skis gelijk moet zijn aan de lengte van de leerlingen. Dat gaat natuurlijk niet, maar kan wel worden geoptimaliseerd door te eisen dat de som van de verschillen van de lengtes zo klein mogelijk moet zijn. Bedenk een greedy algoritme die de skis van lengte h_j zo verdeelt over de leerlingen van lengte ℓ_i dat $\sum |\ell_i - h_j|$ minimaal is (leerling met lengte ℓ_i krijgt skis van hoogte h_j). Wat is de complexiteit van uw algoritme?
3. Er zijn twee algoritmen behandeld om de min cost spanning tree van een gewogen graaf te vinden, die van Prim en die van Kruskal.
 - (a) Werkt één of meer van die algoritmen werken ook met negatieve gewichten?
 - (b) Kan één of meer van die algoritmen gebruikt worden om een spanning tree van *maximum* gewicht te vinden?
4. Bedenk een efficiënte greedy algoritme die een mincost spanning tree vindt door van elke cykel steeds een kant weg te halen.
 - (a) Welke kant moet uw algoritme uit elke cykel halen?
 - (b) Waarom kan een algoritme die eerst *alle* cyclen in een graaf vindt en dan van elke cykel een zware kant weghaalt niet zo'n algoritme zijn?
 - (c) Is de volgorde waarin de cyclen gevonden worden belangrijk?