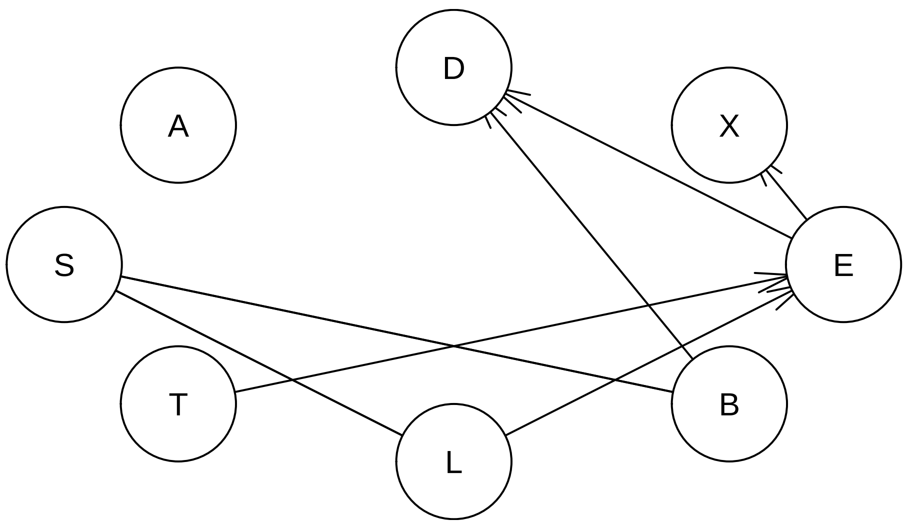
Lab_1

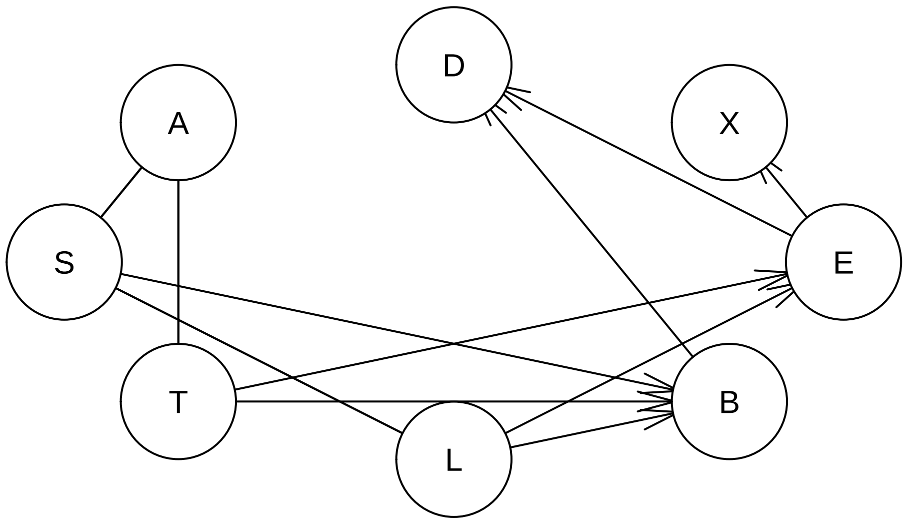
Axel Holmberg (axeho681)

Task 1

Show that multiple runs of the hill-climbing algorithm can return non-equivalent Bayesian network (BN) structures. Explain why this happens. Use the Asia dataset which is in- cluded in the bnlearn package. To load the data, run data("asia").



Above is the junction tree for the data with 5 restarts and an initinital graph based on empty.graph(c("A", "S", "T", "L", "B", "E", "X", "D")).



Above is a junction tree for the data where the score is set by Akaike Information Criterion score(aic) instead of the default Bayesian Information Criterion.

```
## [1] "Different number of directed/undirected arcs"
```

As one can see in the print statement above, as well as in the plots, the hill-climbing algorithm can return non-equivalent BN-strucvtures. The reason for this is that . . . .

Task 2

Learn a BN from 80 % of the Asia dataset. The dataset is included in the bnlearn package. To load the data, run data("asia"). Learn both the structure and the parameters. Use any learning algorithm and settings that you consider appropriate. Use the BN learned to classify the remaining 20 % of the Asia dataset in two classes: S = yes and S = no. In other words, compute the posterior probability distribution of S for each case and classify it in the most likely class. To do so, you have to use exact or approximate inference with the help of the bnlearn and gRain packages, i.e. you are not allowed to use functions such as predict. Report the confusion matrix, i.e. true/false positives/negatives. Compare your results with those of the true Asia BN

```
##
## bn_pred  no yes
##     no  337 121
##     yes 176 366
```

```
##
## bn_pred_true  no yes
##          no  337 121
##          yes 176 366
```